

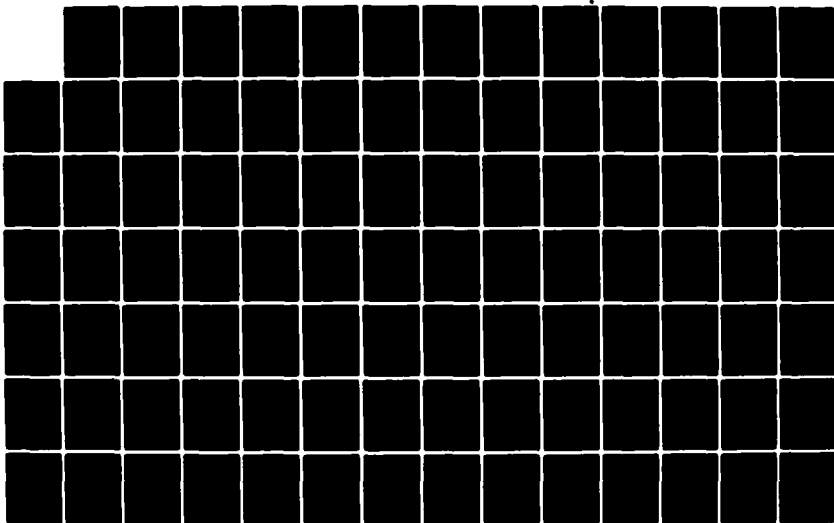
AD-A126 907 NUMERICAL INTEGRATION OF NONEQUILIBRIUM INTERNAL FLOW
USING UNSTEADY EULER EQUATIONS(U) AIR FORCE INST OF
TECH WRIGHT-PATTERSON AFB OH M J TROUT FEB 83

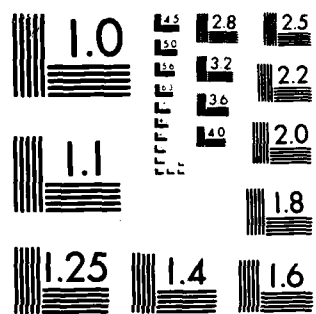
1/2

UNCLASSIFIED AFIT/CI/NR-83-8T

F/G 20/4

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 83-8T	2. GOVT ACCESSION NO. AD-A126907	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Numerical Integration of Nonequilibrium Internal Flow Using Unsteady Euler Equations		5. TYPE OF REPORT & PERIOD COVERED THESIS/DISSERTATION
7. AUTHOR(s) Martin John Trout		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: Massachusetts Institute of Technology		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS FIT/NR PAFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
12. REPORT DATE Feb 83		13. NUMBER OF PAGES 126
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASS
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

SUPPLEMENTARY NOTES

PROVED FOR PUBLIC RELEASE: IAW AFR 190-17

24 Mar 83

Lynn E. Wolaver
 LYNN E. WOLAVER
 Dean for Research and
 Professional Development
 AFIT, Wright-Patterson AFB OH

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

DTIC
 ELECTE
 APR 19 1983
 S E D

DD FORM 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASS

83

04

19

012

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DTIC FILE COPY

AD A 126907

NUMERICAL INTEGRATION OF
NONEQUILIBRIUM INTERNAL FLOW USING
UNSTEADY EULER EQUATIONS

by

Martin John Trout

B.S., Purdue University
(1979)

Submitted in Partial Fulfillment
of the Requirements for the
Degree of

MASTER OF SCIENCE IN
AERONAUTICS AND ASTRONAUTICS

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1983

© Martin J. Trout



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Special
A	

Signature of Author

Martin J. Trout
Department of Aeronautics and Astronautics
January 28, 1983

Certified by

Judson R. Baron

Judson R. Baron
Thesis Supervisor

Accepted by

Harold Y. Wachman

Harold Y. Wachman
Chairman, Department Graduate Committee

NUMERICAL INTEGRATION OF NONEQUILIBRIUM
INTERNAL FLOW USING UNSTEADY
EULER EQUATIONS

by

Martin John Trout

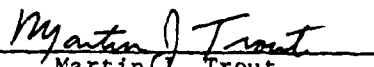
Submitted to the Department of Aeronautics and Astronautics on
January 28, 1983 in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in Aeronautics and Astronautics

ABSTRACT

A quasi-one dimensional, nonequilibrium, streamtube model and finite difference computer code has been developed to study nonisentropic supercritical, real gas flows. A system of unsteady Euler equations is integrated to a steady state solution utilizing MacCormack's explicit/implicit numerical scheme. Implicit characteristic boundary conditions are employed at the subsonic inlet to enable automatic solution of inlet conditions consistent with the critical mass flow rate. A numerical damping technique for the explicitly evaluated reaction rate has been developed such that nonequilibrium flows with large reaction rate coefficients have been solved numerically with significant reduction in computation time.

Thesis Supervisor: Professor Judson R. Baron
Title: Professor of Aeronautics and Astronautics

All rights to reproduce any or all of this thesis are granted to the Massachusetts Institute of Technology.


Martin J. Trout

ACKNOWLEDGEMENTS

The author would like to express deep appreciation to the individuals who provided support and guidance throughout this investigation.

Many thanks to Professor Baron for his endless hours of advice, encouragement and instruction. Without his personal involvement, the completion of this thesis would not have been possible. To my office mates, thank you for making the long hours not seem so long. And finally, I am deeply grateful to my wife, Tylene, and sons Christopher and Philip for their patience and understanding during the course of this research.

TABLE OF CONTENTS

	<u>Page</u>
Abstract	2
Acknowledgements	3
Table of Contents	4
Nomenclature	6
1.0 Introduction and Literature Survey	8
2.0 Analytical and Physical Model and Governing Equations	16
2.1 Chemistry/Thermodynamics	17
2.2 Nonequilibrium Flow	21
2.3 Equilibrium Flow	22
2.4 Time Scales	23
2.5 Boundary and Initial Conditions	24
3.0 Numerical Analysis	26
3.1 Stiffness/Stability Analysis	26
3.2 Explicit/Implicit Integration Scheme	27
3.2.1 Model Equation	28
3.2.2 Euler System	31
3.3 Boundary Conditions	34
4.0 Numerical Solutions	36
4.1 General Description	36
4.2 Numerical Validation	37
4.3 Integration Difficulties	39
4.4 Results	42
4.4.1 Case 1 - Hot Gas Explicit Solution (CFL = 0.9)	44
4.4.2 Case 2 - Hot Gas Explicit/Implicit Solution (CFL = 5.0)	45

	<u>Page</u>
4.4.3 Case 3 - Hot Gas Explicit/Implicit Solution (CFL = 10.0)	45
4.4.4 Comparison of Computation Time	46
4.4.5 Cases 4 and 5 - Cold Gas	48
4.4.6 Boundary Conditions	48
5.0 Conclusions and Recommendations	49
References	51
Tables	53
Figures	58
Appendix A. Eigenvalues, Eigenvectors and Boundary Conditions	96
Appendix B. Computer Program Input Variable Names	104
Appendix C. Program Subroutine Description	108
Appendix D. Computer Program Listing	109

NOMENCLATURE

A	streamtube cross-section area, also symbol for atomic species
a	speed of sound
c_v	frozen specific heat at constant volume
e	specific internal energy
h	specific enthalpy, also Plank's constant
K	cut off coefficient
K_c	equilibrium constant
$K_{f,1}$	forward rate constant for reaction $A_2 + A \rightleftharpoons 2A + A$
$K_{f,2}$	forward rate constant for reaction $A_2 + A_2 \rightleftharpoons 2A + A_2$
k	Boltzmann's constant
L	characteristic length
M_A	molecular weight for atomic species
\dot{m}	mass flow rate
p	pressure
Q	partition function
R	specific gas constant of the mixture
S	entropy
T	temperature
t	time
u	velocity in the streamwise direction
V	volume
\dot{w}	atomic species equation source term
x	streamwise coordinate
y	cross-section coordinate
α	mass fraction of atomic species

β	proportionality constant for integration step size based on chemical relaxation time
γ	ratio of specific heats
θ	characteristic temperature
ρ	density
τ	characteristic time
ϕ	reaction rate coefficient

Subscripts

c	critical
d	dissociation
el	electronic
F	frozen
i	x-mesh point index
rot	rotational
t	throat
tr	translational
vib	vibrational
α	differentiation with respect to α
ρ	differentiation with respect to ρ
0	stagnation

Superscripts

n	time level index
n_f	temperature exponent
*	equilibrium

Chapter 1

INTRODUCTION AND LITERATURE SURVEY

In the past few decades interest in high temperature aerodynamics has increased dramatically. The two primary areas of interest have been the flow downstream of a shock wave (i.e. within a shock layer) and internal expanding flows. The former is of practical interest when designing high speed aircraft or reentry vehicles and the latter is of interest mainly in propulsive devices. Nonequilibrium, expanding, quasi-one dimensional streamtube flow is the focus of this investigation.

A basic phenomenon associated with expanding compressible flows at high temperature and low pressures is the freezing or trapping of energy in chemical or internal energy modes. Hence, energy is withheld from the system. Departure from equilibrium results with the frozen state extreme clearly containing an excess of energy associated with individual particles over that which it would have in an equilibrium state at the local temperature and pressure. This implies a further reduction in temperature which in turn effects the nonequilibrium rate. Since convergent-divergent nozzles or streamtubes effectively convert internal energy into kinetic energy, which in propulsion applications directly translates into thrust, nonequilibrium effects may degrade nozzle performance appreciably.

Some insight into the trends for convergent-divergent nozzle behavior follows on examining the classical quasi-one dimensional, inviscid, compressible equations such as derived in reference [1]. For supercritical (i.e. a transition from subsonic to supersonic) flow, the critical section is at the geometric minimum of the one dimensional nozzle. Downstream of

the critical point both temperature and density decrease rapidly with increase in gas velocity. Typical cooling rates are of the order 10^5 to 10^6 K/sec [2]. Since changes in temperature affect the equilibrium energy distribution curve (i.e. Maxwell-Boltzmann distribution) and the physical mechanisms to redistribute the energy occur at a finite rate, noticeable departures from physical equilibrium are possible.

However, nonequilibrium processes are nonisentropic, hence the perfect gas descriptions for inviscid compressible flow (e.g., Liepmann and Roshko [1]) require modification. For nonisentropic processes the local thermodynamic state is not only a function of streamtube area, but also the upstream area distribution, i.e. the history of the fluid flow upstream of any point in the streamtube. The specific nonequilibrium effects present in a particular flow field depend on the gas components and the temperature and density of the flow. This follows since all such physical phenomena need particle interactions to maintain equilibrium, and temperature and density levels indicate the energy and number of such interactions. Representative types of possible nonequilibrium physical phenomena to be found in an expanding flow are: translational, rotational, vibrational, chemical, and radiation. For any energy mode to maintain local equilibrium, the characteristic time for readjustment by collisions must be negligible compared with the time required for the fluid to experience a significant change in local conditions (characteristic flow time). In general, the adjustment of translational and rotational modes requires relatively few collisions to maintain equilibrium as compared to vibration or a chemical reaction. Hence, translation and rotation have short characteristic times for readjustment by collisions and thus are generally assumed to be in local equilibrium. It is only in the presence of large

flow field gradients that the characteristic readjustment time becomes of the same order as the characteristic flow time. This paper emphasizes nonequilibrium due to chemical reactions, assuming all other modes to be in local equilibrium. However, the methodology is applicable to a general nonequilibrium process which may be modeled and evaluated using the same numerical procedures.

Since chemical reactions occur as a direct result of molecular collisions, it is apparent that as the temperature and density decrease in an expanding flow the number of interactions decrease and energy may effectively become frozen due to "incomplete" chemical reaction. If we consider the simple case of an elementary reaction involving diatomic dissociation, the atomic species concentration (α) is a measure of the energy required to break the diatomic bonds. Such "tied up" energy is then unavailable for conversion into kinetic energy. In a flow process this manifests itself by several distinct regions which are associated with the freezing phenomenon. In subsonic regions, the characteristic flow time is large since the temperature and density are high, and the atomic concentration closely approximates equilibrium distribution. As a flow accelerates in a subsonic section, the concentration departs from equilibrium but still maintains near equilibrium conditions. The flow may then continue to accelerate with corresponding decreases in temperature and pressure. At some point in the streamtube such decreases imply that the characteristic reaction time becomes much larger than the characteristic flow time, since sufficient particle collisions to maintain equilibrium are no longer present. When this occurs, the concentration quite rapidly departs from equilibrium and effectively "freezes out", essentially withholding energy from the flow.

From a practical standpoint, a nozzle, for example, can be designed so as to minimize nonequilibrium effects. There are basically four major parameters of interest here that control nozzle performance and they will be discussed in some detail later. Briefly, they are nozzle scale, reaction rate constant, stagnation density, and stagnation temperature. Previous studies indicate [3] that order of magnitude changes in a nondimensional rate parameter (ϕ) are needed to significantly alter the nonequilibrium concentration. Increasing the stagnation pressure (p_0) does have an effect on both the initial dissociation and the recombination rate. However, for propulsion applications there will be a weight penalty for increases in the stagnation condition. For this reason predicting thrust loss due to nonequilibrium effects is important to the performance of propulsive devices.

Since the nonequilibrium system of equations is nonisentropic, virtually all problems of practical interest are too complicated to solve analytically. Therefore, various schemes have been proposed to numerically integrate the system of partial differential equations. The numerical techniques have primarily been related to the difficulties with: 1) boundary conditions and, 2) near equilibrium situations. One of the earliest methods [3] employed a standard numerical integration technique (Runge-Kutta) applied to a system of steady Euler equations. For an initial value type problems (e.g., a supersonic inlet) the inlet conditions were specified and a solution computed by a space marching method. However, if one boundary is upstream of the critical station, use of a steady Euler equation description implies a two point boundary value problem involving the critical mass flow (\dot{m}_c) and the inlet boundary conditions. The critical mass flow is determined at the station where the

local flow velocity is equal to the local frozen speed of sound. Unlike the isentropic case, (\dot{m}_c) cannot be determined a priori because the flow at each point depends on the entire upstream history. In addition, for nonequilibrium flow the critical point is downstream of the geometric minimum. Therefore, the critical mass flow rate and location must be determined during the integration procedure. This problem is illustrated by the following relation [2]:

$$(1) \quad \frac{1}{u} \frac{du}{dx} = \frac{\left(\frac{1}{A} \frac{dA}{dx} - \frac{h_x}{\rho h_p} \frac{d\alpha}{dx} \right)}{\left(\frac{u^2}{a_F^2} - 1 \right)}$$

which is the nonequilibrium extension of the classical expression [1]:

$$(2) \quad \frac{1}{u} \frac{du}{dx} = \frac{\frac{1}{A} \frac{dA}{dx}}{\frac{u^2}{(dp/dx)/(d\rho/dx)} - 1}$$

for isentropic flow. For $\frac{dA}{dx} = 0$ (geometric minimum) the local velocity equals the local sound speed (a_F or a^*) for isentropic supercritical flow. However, equation (1) indicates the critical point can occur downstream of the throat since $\left(\frac{h_x}{\rho h_p} \right) \left(\frac{d\alpha}{dx} \right)$ is generally positive. This is true because h_p is negative, h_x is positive (except at very high temperatures), and in an accelerating flow $\frac{d\alpha}{dx}$ is negative due to decreasing temperature. At such a downstream critical station the velocity is equal to the local frozen sound speed which is unknown initially.

Several methods have been devised for the two point boundary value problem when using steady state equations. A simple procedure consists of essentially several guesses for the mass flow rate, using the limiting

frozen and equilibrium mass flow rates as guides for upper and lower bounds. Successive iteration were computed until supercritical flow downstream of the throat was calculated (Bray [3], Hall and Russo [4]). A second technique involved integration to just upstream of the critical singularity with a guess for \dot{m}_c . The solution was matched with a series expansion about the singular point to obtain a corrected value for \dot{m}_c , and the procedure was repeated until convergence resulted [3]. An inverse method proposed by Eschenroeder et al [3] eliminated the necessity for iteration by specifying the equilibrium density (ρ^*) instead of the area distribution in the transonic nozzle section. Equilibrium mass flow (\dot{m}_c^*) was assumed and the area ratio ($A(x)$) was determined as part of the solution. Integration was continued downstream of the critical point, then direct integration methods were used for the remainder of the computational domain. Still another technique (Bray [3]) suggested patching together an upstream equilibrium solution to a nonequilibrium flow downstream of the critical point.

A major difficulty in the integration procedure is that typically the upstream boundary point includes a subsonic, near equilibrium region. The departure from close to equilibrium states exhibits a singular perturbation behavior. This results from the fact that the nonequilibrium parameter (e.g. concentration) alters its state in proportion differences from equilibrium and universally proportional to a characteristic time; in the near equilibrium region this is essentially 0/0, i.e. closely indeterminate. If a ordinary explicit space marching scheme is used, extremely small increments must be taken to maintain numerical stability and is somewhat impractical. In reference [5], Lomax and Bailey provide an excellent summary of numerical integration methods applied to the singular

perturbation problem. One may use small perturbation theory in the near equilibrium region to increase step size [6]. Another technique developed by Treanor [7] uses a linear approximation to the rate equation. The linear equation is integrated explicitly with the results being expressed in the form of the Runge-Kutta finite difference equation plus a correction term. Step size can also be increased by converting the differential equations to integral equations. Bray [3] avoids the problem by using his patched equilibrium solution technique. Finally implicit numerical techniques are suggested in reference [5], and a review of implicit methods as applied to reacting systems is given in reference [8]. Implicit schemes do remove the step size constraint of explicit schemes but are less accurate and require more computer time per iteration.

In recent years consideration has been given to circumventing the two point boundary value problem by means of an unsteady description. The system of equations is then hyperbolic in time throughout the entire domain of integration in contrast to the elliptic subsonic region when steady. The technique has been applied to ideal compressible [9] and nonequilibrium [10] flows using explicit integration schemes (e.g., Lax-Wendroff [11] or MacCormack [12]). Unsteady methods allow integration in time to a steady state compatible with imposed steady state boundary conditions. The critical mass flow and critical section location are then found in the course of the numerical evaluation. However, the unsteady scheme still must converge to a steady state solution in all near equilibrium regions, and this limits step size in much the same way as when using steady Euler equations.

It is the present intent to take advantage of both time dependent Euler equations, which eliminate the two point boundary value problem, and

implicit schemes which eliminate the stability time step constraint. In Chapter 2, the physical model for nonequilibrium, compressible flow in a streamtube will be described, including assumptions and model equations. Boundary conditions, thermodynamics, and rate chemistry are also discussed. In Chapter 3, the suggested implicit numerical scheme is described and numerical treatment of the boundaries is discussed. Chapter 4 covers numerical solutions that illustrate the capability of the implicit scheme when applied to nonequilibrium flow. Finally, Chapter 5 discusses possible improvements, extensions to higher dimensions and larger systems of reactions.

Chapter 2

ANALYTICAL AND PHYSICAL MODEL AND GOVERNING EQUATIONS

The modeling is that of a nonequilibrium flow in a quasi-one dimensional streamtube. If all processes were isentropic, algebraic equations would describe the thermodynamic changes of state [13]. Thermodynamics then constrains the inter-related state variables but implies a one to one correspondence between physical and dynamic properties calculated from the equations of motion. For nonequilibrium flow the thermodynamic behavior depends on the detailed flow history. As mentioned, the unsteady Euler equations will be used to model the nonequilibrium flow. They comprise a hyperbolic system of partial differential equations which describe an inviscid, adiabatic, compressible flow. The quasi-one dimensional system is then:

$$\text{mass: } \frac{\partial \rho}{\partial t} + \frac{1}{A} \frac{\partial (\rho u A)}{\partial x} = 0$$

$$\text{momentum: } \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = -\frac{1}{\rho} \frac{\partial p}{\partial x}$$

(3)

$$\text{energy: } \frac{\partial e}{\partial t} + u \frac{\partial e}{\partial x} = -\frac{p}{\rho} \frac{\partial \rho}{\partial x} - \frac{p u}{\rho A} \frac{\partial A}{\partial x}$$

$$\text{species: } \frac{\partial \alpha}{\partial t} + u \frac{\partial \alpha}{\partial x} = \dot{\omega}(\rho, T, \alpha)$$

In addition to the standard (homogeneous medium) conservation equations, there is a single conservation of species constraint. The physical modeling required by this phenomenon is presented in the following section.

2.1 Chemistry/Thermodynamics

The species equation, in fact, is representative of a number of possible nonequilibrium effects: for example, internal modes such as vibrational, rotational, or electronic, and chemical modes such as dissociation, ignition. Here we will assume that chemical dissociation is representative and the sole nonequilibrium flow phenomenon. This implies that all other energy modes are either in equilibrium at the local flow temperature (T) or maintain a fixed sub-state, i.e. frozen. The species concentration effects the system both implicitly and explicitly. That is, pressure and internal energy are now functions of concentration as well as two other thermodynamic properties, such as density and temperature.

A relatively simple binary gas suffices to illustrate the nonequilibrium effects. An approximate but realistic gas model described by Lighthill/Freeman [13], [14] as an "ideal dissociating gas," and contains the major feature of a binary reacting system while simultaneously eliminating complex but non-enlightening terms. Lighthill simplified several contributions to the partition functions for a diatomic gas. A brief discussion of the basis is described below. Consider a diatomic dissociating gas to be composed of particles (A) which include translational and electronic internal energy modes, and a combined particle (A_2) which, in addition to the above, has vibrational and rotational degrees of freedom. The factorized partition functions of each are of the form

$$\begin{aligned} Q^A &= Q_{tr}^A Q_{el}^A \\ (4) \quad Q^A &= Q_{tr}^{A_2} Q_{rot}^{A_2} Q_{vib}^{A_2} Q_{el}^{A_2} \end{aligned}$$

The translational partition function for (A) atoms or (A_2) molecules is

$$(5) \quad Q_{tr} = V \left(\frac{2\pi m k T}{h^2} \right)^{3/2}$$

where m is the mass of the specific particle. The rotational partition function is

$$(6) \quad Q_{rot}^{A_2} = \frac{1}{2} \left(\frac{T}{\Theta_r} \right)$$

and the vibrational partition function is given by

$$(7) \quad Q_{vib}^{A_2} = \frac{1}{1 - e^{-\Theta_v/T}}$$

The electronic partition function is described by the general equation

$$(8) \quad Q_{el} = g_0 e^{-\epsilon_0/kT} + g_1 e^{-\epsilon_1/kT} + g_2 e^{-\epsilon_2/kT} + \dots$$

where the g_i are degeneracy factors for the energy levels ϵ_i . Specific values for the constants for a particular species are obtained by spectroscopic analysis of transition between energy levels. Use of equations (5) - (8) in the mass action equation for a symmetric diatomic gas leads to:

$$(9) \quad \frac{\alpha^{*2}}{1 - \alpha^{*2}} = \frac{e^{-\Theta_d/T}}{P} \left[m \left(\frac{\pi m k}{h^2} \right)^{3/2} \Theta_r \sqrt{T} (1 - e^{-\Theta_v/T}) \frac{(Q_{el}^A)^2}{(Q_{el}^{A_2})^2} \right]$$

where (α^*) indicates equilibrium concentration. Lighthill noted that for $1000 < T < 7000^\circ\text{K}$ the contents of the brackets in equation (9) is essentially constant, and he suggested that it be taken to be a constant, characteristic, dissociation density (ρ_d). The simplified expression is then

$$(10) \quad \frac{\alpha^*{}^2}{1 - \alpha^*} = \frac{p_d}{p} e^{-\Theta_d/T}$$

The thermal equation of state is unaffected by the constant p_d assumption since pressure depends only on derivatives of logarithms of the partition function with respect to volume. Assuming both A and A_2 to be perfect gases, the thermal equation of state in equilibrium is

$$(11) \quad p = (1 + \alpha^*) R_{A_2} p T$$

The expression $(1 + \alpha^*) R_{A_2}$ is essentially the equivalent of a mixture gas constant.

Since the caloric equation is a function of temperature it is affected by the Lighthill approximation. The general expression for equilibrium is [13]:

$$(12) \quad e^* = R_{A_2} T^2 \left[2\alpha^* \frac{\partial}{\partial T} \ln Q^A + (1 - \alpha^*) \frac{\partial}{\partial T} \ln Q^{A_2} \right] - (1 - \alpha^*) R_{A_2} \Theta_d$$

Using the simplified mass action relation (10), this reduces to simply:

$$(13) \quad e^* = R_{A_2} \left[3T + \alpha^* \Theta_d \right]$$

in which contributions due to electronic energy have been assumed to be negligible for the temperature range considered. The internal energies of individual constituents of the mixture are:

$$(14) \quad e_A = (e_A)_{tr} = \frac{3}{2} R_A T = 3 R_{A_2} T$$

Neglecting electronic energy and on assuming that the molecular vibration mode is half excited:

$$e_{A_2} = \frac{3}{2} R_{A_2} T + R_{A_2} T + \frac{1}{2} R_{A_2} T = 3 R_{A_2} T$$

Since $e = \alpha e_A + (1-\alpha)e_{A_2}$ Eq. (13) then follows. Note that in this model the ratio of specific heats for a non-dissociated gas ($\alpha = 0$) is $\frac{4}{3}$, which is less than the true $\frac{7}{5}$ for diatomic molecules. From Eqs. (11) and (13) the equilibrium enthalpy is:

$$(15) \quad h \equiv e + \frac{p}{\rho} = R_{A_2} \left[(4 + \alpha^*)T + \alpha^* \Theta_d \right]$$

The preceding discussion of equilibrium chemistry provides a basis for the nonequilibrium model. The thermal (11) and caloric (13) equations of state are of the same form for nonequilibrium and require only replacement of α^* by α . A finite rate associated with concentration changes requires replacement of the algebraic mass action equation by a differential description equation (3d). The right hand side of this equation represents the production or recombination of atoms. For a binary diatomic dissociating gas

$$(16) \quad \dot{w}(\rho, T, \alpha) = \left(k_{f,1} \alpha + k_{f,2} \frac{1-\alpha}{2} \right) \frac{\rho}{M_A} \left[(1-\alpha) - \frac{2\rho}{M_A K_C} \alpha^2 \right]$$

Freeman [13] noted that for equilibrium

$$\frac{\alpha^{*2}}{1-\alpha^*} = \frac{M_A K_C}{2\rho}$$

which is identical to the ideal model equation (10) if $M_A \frac{K_C}{2\rho}$ is replaced by $\frac{p_d}{\rho} e^{-\Theta_d/T}$. He also replaced the first parenthetical factor on the right hand side of equation (16) by a single Arrhenius rate constant.

$$(17) \quad \left(k_{f,1} \alpha + k_{f,2} \frac{(1-\alpha)}{2} \right) = C_f T^{\eta_f} e^{-\Theta_f/T}$$

Therefore the source term becomes

$$(18) \quad \dot{w} = C_f T^{\eta_f} \rho \left[(1-\alpha) e^{-\Theta_d/T} - \frac{\rho}{\rho_d} \alpha^2 \right]$$

2.2 Nonequilibrium Flow

With the motion described by unsteady Euler equations and thermodynamics/chemistry described by an ideal dissociating gas model, the following system of nondimensional equations apply:

$$(19) \quad U_t + A U_x + H = 0$$

where:

$$U = \begin{bmatrix} p \\ u \\ T \\ \alpha \end{bmatrix}, \quad A = \begin{bmatrix} u & p & 0 & 0 \\ \frac{RT}{\rho_0 b p} u & \frac{p}{\rho_0 b} & \frac{R_u T}{\rho_0 b} \\ 0 & RT/c_v & u & 0 \\ 0 & 0 & 0 & u \end{bmatrix}, \quad H = \begin{bmatrix} \frac{\rho u}{A} \frac{\partial A}{\partial x} \\ 0 \\ \frac{RT_u}{c_v A} \frac{\partial A}{\partial x} + q \\ - \dot{w} \end{bmatrix}$$

The nondimensional parameters are:

$$(20) \quad \begin{aligned} p &= p'/p_0, & u &= u'/a_0, & T &= T'/T_0 \\ x &= x'/L, & t &= t'/(L/a_0), & A &= A'/A_t \end{aligned}$$

where ()' indicates the dimensional variables in (3) and ()₀ indicates stagnation conditions.

$$\dot{w} = \Phi T^{n_d} p \left[(1-\alpha) e^{-\Theta_d/T} - \frac{p}{p_d} \alpha^2 \right]$$

$$\Phi = (L/a_0) / (M_A / c_v T_0^{n_d} p_0)$$

$$(21) \quad R = (1 + \alpha) R_{A_2}$$

$$c_v = 3 R_{A_2}$$

$$q = \frac{\dot{w}}{T_0 c_v} (e_A - e_{A_2}) = \frac{\Theta_d R_{A_2}}{T_0 c_v} \dot{w}$$

2.3 Equilibrium Flow

A limiting case to be considered is that of idealized "identical" equilibrium. In this limit the differential species conservation equation may be replaced by the equivalent law of mass action (10). The order of the equation system decreases by one and local equilibrium is enforced instantaneously and continuously. Thus, the state vector and coefficient matrices of Eq. (19) are now

$$\begin{aligned}
 \mathbf{U}^* &= \begin{bmatrix} \rho \\ u \\ T \end{bmatrix}, & \mathbf{H}^* &= \begin{bmatrix} \frac{\rho u}{A} \frac{\partial A}{\partial x} \\ 0 \\ k_1 \left(\frac{RT}{C_v} - k_2 \rho \right) \frac{u}{A} \frac{\partial A}{\partial x} \end{bmatrix} \\
 (22) \quad \mathbf{A}^* &= \begin{bmatrix} u & \rho & 0 \\ \left(\frac{R_{A_2} T}{\gamma_0 R_0} \frac{\partial \alpha}{\partial \rho} + \frac{R^* T}{\gamma_0 R_0 c} \right) & u \left(\frac{R^*}{\gamma_0 R_0} + \frac{R_{A_2} T}{\gamma_0 R_0} \frac{\partial \alpha}{\partial T} \right) & \\ 0 & k_1 \left(\frac{RT}{C_v} - k_2 \rho \right) & u \end{bmatrix}
 \end{aligned}$$

where:

$$\begin{aligned}
 k_1 &= \frac{1}{1 + \frac{\Theta_d}{3T_0} \frac{\partial \alpha}{\partial T}}, & k_2 &= \frac{\Theta_d}{3T_0} \frac{\partial \alpha}{\partial \rho} \\
 (23) \quad R^* &= (1 + \alpha^*) R_{A_2}, & \frac{\alpha^{*2}}{1 - \alpha^{*2}} &= \frac{p_d}{p} e^{-\alpha_d/T}
 \end{aligned}$$

Actually, for the isentropic frozen or equilibrium limits, the entire system can be reduced to algebraic forms. However, for consistency such limiting cases have been integrated here using the same unsteady Euler equation description.

2.4 Time Scales

When the flow is neither frozen ($\alpha = \text{constant}$) nor in equilibrium, there are two time scales present in the problem. Multiple time scales can lead to integration difficulties (see Chapter 3). Such problems are strictly numeric and result from the presence of widely separated eigenvalues in the discrete equations. The two scales are a flow time (τ_{flow}) and a chemical relaxation time (τ_{chem}). The flow time

$$(24) \quad \tau_{\text{flow}} = L/a_0$$

is a measure of the physical scale of the nozzle or residence time of the fluid as was suggested in Eq. (20). The chemical relaxation time

$$(25) \quad \tau_{\text{chem}} = \frac{M_A}{C_f p_0 T_0^{\eta_f}}$$

is a measure of how rapidly the fluid will relax to an equilibrium state. The rate parameter ϕ in Eq. (21) is simply the ratio of the time scales, $\tau_{\text{flow}}/\tau_{\text{chem}}$.

This process can be understood in terms of a linearized model for a rate equation; say

$$(26) \quad \frac{d\alpha}{dt} = \frac{\alpha^* - \alpha}{\tau}$$

in which τ is a modified τ_{chem} .

In general α^* and τ are both time dependent, but for constant values the decay towards equilibrium exhibits a simple exponential behavior (Fig. 1):

$$(27) \quad \frac{\alpha - \alpha^*}{\alpha_0 - \alpha^*} = e^{-t/\tau}$$

Clearly τ is the time required to decrease the relative difference by the factor e^{-1} , and smaller τ implies a faster relaxation process. Of course the assumed constant heat bath in general, is not true along a streamtube. Thus variable temperature and density imply varying α^* and τ_{chem} according

to local conditions, and hence there is a coupling to the fluid motion.

The limit $\frac{\tau_{\text{flow}}}{\tau_{\text{chem}}} \ll 1$ is the frozen case in which there is no time to react

to local flow changes; alternatively, this is equivalent to a short path

length. The opposite extreme, $\frac{\tau_{\text{flow}}}{\tau_{\text{chem}}} \gg 1$, approximates local equilibrium

and corresponds to a relatively long local residence time for a particle.

It is the near equilibrium situation for which neither time scale can be neglected, but the chemical relaxation time is relatively small, that lead to numerical difficulties.

2.5 Boundary and Initial Conditions

The boundary conditions are extremely important and require special care in the case of numerical evaluations. For a subsonic inlet two boundary conditions are required, and are taken here to be stagnation enthalpy and entropy. Since interest is in the steady state, only steady state boundary conditions are applied. The total enthalpy is a function of temperature, velocity and concentration:

$$(28) \quad h_0 = R_{A_2} \left[(4 + \alpha) T + \alpha \theta_d \right] + \frac{u^2}{2}$$

The differential entropy is [15]

$$(29) \quad ds = 3R_{A_2} \frac{dT}{T} - (1 + \alpha) R_{A_2} \frac{dp}{p} - R_{A_2} \ln \left[\frac{\alpha^2}{\alpha^{*2}} \frac{1 - \alpha^*}{1 - \alpha} \right] d\alpha$$

The limiting expressions for frozen and equilibrium extremes are [13]

$$(30) \quad \begin{array}{l} \text{Equilibrium:} \\ S = 3 \ln \frac{I}{\theta_d} + \alpha^* (1 - 2 \ln \alpha^*) - (1 - \alpha^*) \ln (1 - \alpha^*) \\ \quad \quad \quad - (1 + \alpha^*) \ln \frac{p}{p_d} + \text{const} \\ \text{Frozen:} \\ \frac{I}{p^{\gamma-1}} = \text{Const} \end{array}$$

More will be said regarding boundary and initial conditions (including the supersonic exit) in Chapter 3 where the specific application to the numerical technique is discussed.

Chapter 3

NUMERICAL ANALYSIS

3.1 Stiffness/Stability Analysis

It has been pointed out that the nonequilibrium system of equation (19) is numerically stiff. Stiffness is not the result of an analytical technique but can originate from the discretization of a continuum equation set when multiple time scales (e.g. τ_{flow} and τ_{chem}) are present. The phenomenon can be demonstrated by writing the species continuity equation in finite difference form using explicit upwind differencing ($u > 0$):

$$(31) \quad \alpha_i^{n+1} = \alpha_i^n - c(u_i^n - u_{i-1}^n) - d(\alpha_i^n - \alpha_i^0)$$

$$\text{where } c = \frac{u \Delta t}{\Delta x} \quad (\text{Courant number})$$

$$d = \frac{\Delta t}{\tau_{\text{chem}}}$$

The last term in equation (31) represents the time linearized form of the chemical rate equation [13]. The stability limits of this explicit finite difference algorithm can be investigated using Von Neumann stability analysis [16]. Each Fourier component of the solution is written as:

$$(32) \quad \alpha_i^n = V^n e^{I i \theta}$$

which on substituting into (31) yields

$$(33) \quad \frac{V^{n+1}}{V^n} = 1 - c(1 - e^{-I \theta}) - d$$

Since the modulus of the amplification factor must not exceed unity, i.e.

$$(34) \quad |G| \equiv \left| \frac{V^{n+1}}{V^n} \right| \leq 1$$

therefore

$$(35) \quad G^2 = 1 + 2c(1 - \cos \theta)(c + d - 1) + d(d - 2) \leq 1$$

If satisfied for all θ and $0 \leq 1 - \cos \theta \leq 2$

$$(36) \quad 4c(c + d - 1) + d(d - 2) \leq 0$$

or

$$(37) \quad c \leq 1 - d/2$$

The classical upwind constraint follows for the frozen limit ($d \rightarrow 0$), namely

$$(38) \quad c \leq 1$$

However, notice that positive c requires

$$(39) \quad d \leq 2$$

or

$$(40) \quad \Delta t \leq \frac{\tau_{chem}}{\frac{1}{2} + \frac{u \tau_{chem}}{\Delta x}}$$

Therefore, as $\tau_{chem} \rightarrow 0$ (higher reaction rates) the step size (Δt) for stability is determined by d rather than c . Hence, for situation in which step size is dictated by chemistry, the number of iterations for convergence to the steady state may be impractical for explicit techniques. An implicit method is of some interest in order to improve the step size and therefore, iteration count.

3.2 Explicit/Implicit Integration Scheme

The specific technique applied here to the nonequilibrium equation set is that developed by MacCormack [17]. The scheme is an implicit analog of MacCormack's explicit algorithm [12] introduced in 1969. In fact, the new method uses the earlier explicit technique as the first of a two stage procedure and is referred to as an explicit/implicit algorithm. The explicit first stage uses the original predictor - corrector concept to compute local changes in the dependent variables of the governing

equations. Then, the implicit second stage distributes those changes globally to all points in the computational domain, thus removing the stability constraint associated with the explicit algorithm. The procedure is second order accurate in time and space, unconditionally stable in time, and block bidiagonal in form.

3.2.1 Model Equation

The basic theory and implementation of the explicit/implicit algorithm can be demonstrated using a model convection equation:

$$(41) \quad \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$$

The explicit part integrates equation (41) using a predictor/corrector method:

$$(42) \quad \begin{aligned} P: & \begin{cases} \Delta u_i^n = -\frac{\Delta t c}{\Delta x} (u_{i+1}^n - u_i^n) \\ u_i^{\overline{n+1}} = u_i^n + \Delta u_i^n \end{cases} \\ C: & \begin{cases} \Delta u_i^{\overline{n+1}} = -\frac{\Delta t c}{\Delta x} (u_i^{\overline{n+1}} - u_{i-1}^{\overline{n+1}}) \\ u_i^{n+1} = \frac{1}{2} (u_i^n + u_i^{\overline{n+1}} + \Delta u_i^{\overline{n+1}}) \end{cases} \end{aligned}$$

In the predictor step, a known field at time $t = n\Delta t$ is used to provide an initial estimate for the new field at time level $t = (n+1)\Delta t$ by using one sided spatial difference to compute the dependent variable change, Δu , for the hyperbolic equation. The estimates u_i^{n+1} are used in the corrector step with opposite one sided differencing to compute a second estimate of the change Δu_i^{n+1} . The new prediction is then the previous value plus an average of the two estimates. This can be clearly demonstrated by

rewriting the finite difference equations (42) as an Euler predictor followed by a modified Euler corrector [18]:

$$(43) \quad \begin{aligned} u_i^{n+1} &= u_i^n + \Delta t u_i'^n \\ u_i^{n+1} &= u_i^n + \frac{1}{2} \Delta t (u_i'^n + u_i'^{n+1}) \end{aligned}$$

These are equivalent to MacCormack when the time derivatives of a hyperbolic equation (u') are computed using forward spatial differences in the predictor and backward spatial differences in the corrector. Hence, by equation (43) the change in the dependent variable (u) is the average of the changes at the predictor and corrector levels. The method is second order accurate and stable if the time step satisfies the usual CFL condition:

$$\Delta t \leq \Delta x / c$$

Since C in Eq. (41) is the slope of the characteristic in the physical domain, this restriction simply insists that the computational domain of dependence must include the physical domain of dependence (figure 2).

The stability constraint is removed by including both current ($t = n\Delta t$) and new ($t = (n+1)\Delta t$) time levels in the finite difference approximation of equation (41). Along a negative slope ($c < 0$), forward spatial differencing is appropriate since the information is propagating from larger x at earlier time. Therefore an implicit form for equation (41) becomes

$$(44) \quad u_i^{n+1} = u_i^n - (1-\alpha) \frac{c \Delta t}{\Delta x} (u_{i+1}^n - u_i^n) - \alpha \frac{c \Delta t}{\Delta x} (u_{i+1}^{n+1} - u_i^{n+1})$$

The constant (α) essentially determines the "degree of implicitness" in the equation. If ($\alpha = 0$), the original explicit equation is recovered; if ($\alpha = 1$), equation (44) is fully implicit. On collecting terms, Equation

(44) can be written in the following form:

$$(45) \quad \left(1 + \lambda \frac{\Delta t}{\Delta x}\right) \delta u_i^{n+1} = -c \frac{\Delta t}{\Delta x} \Delta_+ u_i^n + \lambda \frac{\Delta t}{\Delta x} \delta u_{i+1}^{n+1}$$

where

$$(46) \quad \begin{aligned} \delta u_i^{n+1} &= u_i^{n+1} - u_i^n \\ \Delta_+ u_i^n &= u_{i+1}^n - u_i^n \\ \lambda &= \alpha |c| \end{aligned}$$

Similarly, if $c > 0$

$$(47) \quad \left(1 + \lambda \frac{\Delta t}{\Delta x}\right) \delta u_i^{n+1} = -c \frac{\Delta t}{\Delta x} \Delta_- u_i^n + \lambda \frac{\Delta t}{\Delta x} \delta u_{i-1}^{n+1}$$

where

$$\Delta_- u_i^n = u_i^n - u_{i-1}^n$$

For stability, it can be shown that

$$(48) \quad \lambda \geq \frac{1}{2} \left(|c| - \frac{\Delta x}{\Delta t} \right)$$

Note that the first term on the right hand side of equation (45) is the explicit approximation (42) to the model equation. Hence, the implicit extension of the original MacCormack explicit scheme is

$$(49a) \quad P: \begin{cases} \Delta u_i^n \text{ explicit} \\ \left(1 + \lambda \frac{\Delta t}{\Delta x}\right) \delta u_i^{\overline{n+1}} = \Delta u_i^n + \lambda \frac{\Delta t}{\Delta x} \delta u_{i+1}^{\overline{n+1}} \\ u_i^{\overline{n+1}} = u_i^n + \delta u_i^{\overline{n+1}} \end{cases}$$

$$(49b) \quad C: \begin{cases} \Delta u_i^{\overline{n+1}} \text{ explicit} \\ \left(1 + \lambda \frac{\Delta t}{\Delta x}\right) \delta u_i^{n+1} = \Delta u_i^{\overline{n+1}} + \lambda \frac{\Delta t}{\Delta x} \delta u_{i-1}^{n+1} \\ u_i^{n+1} = \frac{1}{2} (u_i^n + u_i^{\overline{n+1}} + \delta u_i^{n+1}) \end{cases}$$

This system is unconditionally stable for integration in time if λ satisfies the constraint equation (48). In addition, both predictor and corrector procedures involve bidiagonal matrix equations so that only a single sweep through the computational domain is necessary for each level. The bidiagonal structure can be seen from:

$$\begin{bmatrix} M_1 & N_2 & & & & \\ & M_2 & N_3 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & M_{I-1} & N_I \\ & & & & & M_I \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \delta u_2 \\ \vdots \\ \vdots \\ \delta u_{I-1} \\ \delta u_I \end{bmatrix} = \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \vdots \\ \vdots \\ \Delta u_{I-1} \\ \Delta u_I + N_{I+1} \Delta u_{I+1} \end{bmatrix}$$

where

$$M_i = 1 + \frac{\lambda_i \Delta t}{\Delta x}$$

$$N_i = \frac{\lambda_i \Delta t}{\Delta x}$$

3.2.2 Euler System

Using the basic algorithm, the scheme is applicable to the nonequilibrium Euler System (19) as represented by:

$$U_t + A U_x + H = 0$$

The explicit/implicit algorithm for matrix equation (49) is

$$(50a) \quad P: \begin{cases} \Delta U_i^n = -\Delta t \left(A_i^n \frac{\Delta_+ U_i^n}{\Delta x} + H_i^n \right) \\ \left(I - \Delta t \Delta_+ \frac{|A|_i^n}{\Delta x} \right) \delta U_i^{n+1} = \Delta U_i^n \\ U_i^{n+1} = U_i^n + \delta U_i^{n+1} \end{cases}$$

$$(50b) \quad C: \begin{cases} \Delta U_i^{n+1} = -\Delta t \left(A_i^{n+1} \frac{\Delta_- U_i^{n+1}}{\Delta x} + H_i^{n+1} \right) \\ \left(I + \Delta t \frac{\Delta_- |A|_i^{n+1}}{\Delta x} \right) \delta U_i^{n+1} = \Delta U_i^{n+1} \\ U_i^{n+1} = \frac{1}{2} \left(U_i^n + U_i^{n+1} + \delta U_i^{n+1} \right) \end{cases}$$

where

$$\frac{\Delta_+ |A|_i^n}{\Delta x} = \frac{|A|_{i+1}^n - |A|_i^n}{\Delta x}$$

$$\frac{\Delta_- |A|_i^n}{\Delta x} = \frac{|A|_i^n - |A|_{i-1}^n}{\Delta x}$$

The matrix $|A|$ is related to the eigenvalues of the system by

$$(51) \quad |A| = S_x^{-1} D_A S_x$$

where

$$D_A = \begin{bmatrix} \lambda_{A_1} & & & \\ & \lambda_{A_2} & & \\ & & \lambda_{A_3} & \\ & & & \lambda_{A_4} \end{bmatrix}$$

(52)

$$\lambda_{A_1} = \max \{ |u-a| - \Delta x / \Delta t, 0.0 \}$$

$$\lambda_{A_2} = \max \{ |u| - \Delta x / \Delta t, 0.0 \}$$

$$\lambda_{A_3} = \max \{ |u+a| - \Delta x / \Delta t, 0.0 \}$$

$$\lambda_{A_4} = \max \{ |u| - \Delta x / \Delta t, 0.0 \}$$

The matrix of eigenvectors (S_x) satisfies

$$A = S_x^{-1} \Lambda_A S_x$$

where Λ_A is the diagonal matrix of eigenvalues of (A). (Appendix A provides a derivation of the eigenvalues and eigenvectors for both the equilibrium and nonequilibrium equations). For regions which satisfy the usual one dimensional, compressible, explicit stability criteria

$$(53) \quad \Delta t \leq \frac{\Delta x}{|u| + a}$$

all λ_A will be less than or equal to zero. In such cases the λ_A are set to zero (see equation (52)) and the set of difference equations (50) reduce to an explicit form. When all λ_A are positive the implicit equation can be expanded to examine the matrix block bidiagonal form. For example, the implicit predictor equation (50a) can be written as

$$(54) \quad \left(I + \frac{\Delta t}{\Delta x} |A|_i^n \right) \delta U_i^n = \Delta U_i^n + \frac{\Delta t}{\Delta x} |A|_{i+1}^n \delta U_{i+1}^n$$

and solved by a single mesh sweep in the decreasing (i) direction. The corresponding corrector sweep is in the opposite direction. Appendix A provides a detailed description of the block-bidiagonal solver.

This explicit/implicit method has several significant advantages over other implicit methods. One major advantage is the block-bidiagonal structure. Numerous other implicit schemes (e.g. Beam and Warming [19]) employ a block-tridiagonal solution algorithm which tends to increase computer time per iteration. In addition the bidiagonal method is straightforward to program and does not require matrix inversion.

An area of difficulty is the inclusion of a source term (H) in the implicit stage of integration. For an ideal gas, the quasi-one dimensional source term is not time dependent and therefore, does not effect stability. However, the nonequilibrium source term contains the time dependent reaction rate (\dot{w}). (See equation (19)). As will be shown in Chapter 4,

the implicit stage of integration will provide sufficient stability for moderate reaction rates. But, sufficiently high reaction rates effect stability since \dot{w} is only evaluated during the explicit stage of integration.

3.3 Boundary Conditions

The solution procedure for the block bidiagonal matrix equation (48) assumes that all dependent variables are known at the boundaries. For certain types of boundaries (such as supersonic inlets or outlets) the boundary conditions are straightforward. However, for supercritical nozzle flow the boundary conditions require special attention.

Characteristic theory provides a basis for the construction of stable implicit boundary conditions. As noted in reference [20], a correct formulation of boundary conditions are of extreme importance as they are the governing elements of the computation. Porous boundaries in hyperbolic problems are represented schematically in Figure 3 for one dimensional ideal gas flow. (Reacting flows have one additional characteristic; see Appendix A). At a subsonic inlet, two positive eigenvalues ($\lambda_1 = u$, $\lambda_2 = u+d$) define paths along with information is transmitted to the boundary from outside the integration domain. Consequently, two boundary conditions are needed [21]. Along the remaining negative characteristic, information is transmitted from inside the domain; forward spatial differences then correctly account for the influence of such characteristics at the boundary.

To apply these characteristic constraints to the implicit algorithm, the following time linear approach was used [22]. The boundary conditions

$$(55) \quad B_i (U) = 0$$

Can be written in time linear form:

$$(56) \quad \frac{\partial B_c(U)}{\partial t} = \frac{\partial B_c}{\partial U} \frac{\partial U}{\partial t} = 0$$

Recall (Chapter 3) that the boundary conditions for a subsonic inlet are that total enthalpy and entropy are fixed at the inlet. Equation (49) in characteristic form is

$$(57) \quad P U_t + \Delta P U_x + P H = 0$$

where

$$A = P^{-1} \Delta P$$

At the inlet, the eigenvectors corresponding to the positive eigenvalues are replaced by the boundary conditions (56). Therefore equation (57) becomes

$$(58) \quad P_1 U_t + P_2 A U_x + P_2 H = 0$$

$$\text{where } P_1 = \begin{bmatrix} \vec{p}_1 \\ \frac{\partial h_0}{\partial U} \\ \frac{\partial S}{\partial U} \end{bmatrix}$$

$$P_2 = \begin{bmatrix} \vec{p}_1 \\ 0 \\ 0 \end{bmatrix}$$

or

$$(59) \quad U_t + P_1^{-1} P_2 A U_x + P_1^{-1} P_2 H = 0$$

Equation (19) is a modified partial differential equation to be solved at the boundary using the explicit/implicit method (50). Equation (50) need not be modified at the supersonic outlet since there eigenvalues are positive, implying that all information at this boundary is transmitted outward from the interior of the domain. Thus, the boundary condition procedure involves the use of forward difference at the inlet and backward differences at the outlet. The implementation of this procedure is represented schematically by Figure 4. Notice that the mesh sweep was applied in alternating directions to eliminate oscillations.

Chapter 4

NUMERICAL SOLUTIONS

4.1 General Description

A computer program has been developed from application of the physical modeling and numerical techniques of Chapters 2 and 3 to a quasi-one dimensional, nonequilibrium, supercritical, streamtube flow. Code validation and case studies (operating on a PDP 11/23 computer) are presented here to indicate the utility of the code and some advantages relative to earlier algorithms.

Memory limitations (28 Kilobytes) required a sequentation of the domain in order to achieve larger area ratios. An upstream segment (see Figure 5) contained the streamtube minimum section and required the subsonic inlet boundary condition discussed in Chapter 3. Once supersonic flow velocity was achieved, a fixed supersonic boundary condition was used at the inlet of any additional downstream streamtube sections. This procedure allowed for expansion to an arbitrarily large area ratio without either exceeding the memory limitations of the system or sacrificing accuracy as would be required by an otherwise decrease in the number of mesh points for a given area change.

Initial conditions ($t=0$), boundary conditions (inlet, outlet), and computational methods were explored in the selection of example cases for numerical evaluation. Table 1 indicates the individual parameters and specifications for the study, but only some combinations of the overall possibilities were actually used as a basis for the solutions. A majority of cases were considered with the larger stagnation species concentration (referred to as the "hot gas" case). However, a selected set of results

for the lesser ("cold gas") species concentration are also presented. The area distribution along the streamtube was chosen to be parabolic, the larger area ratio level refers to the exit maximum of any extended supersonic section (Figure 5). Initial conditions are the assumed distributions of dependent physical property variables. Typically, a decreasing (downstream) linear variation was employed in order to be consistent with the trends of expanding flow.

4.2 Numerical Validation

It is informative to compare any results with the corresponding expected trends from analytical considerations such as described in references [1], [13] or [23]. Some important checks for consistency are: constant mass flow rate along the streamtube, achievement of sonic velocity at the minimum section in the limiting frozen or equilibrium calculations, and appropriate shifting of the sonic point to a downstream location for nonequilibrium flow. Of course, accuracy of a numerical result depends upon the number of mesh points in the computational domain, integration step size, and the enforced convergence criteria. The latter was taken here to be

$$(60) \quad \sum_i \left| U^{n+10} - U^n \right| < \epsilon \quad (= 10^{-4})$$

as the criteria for all computed results.

For frozen flow, computed results were compared with tabulated, analytical results, e.g. [23]. Table 2 summarizes the numerical solution of an ideal gas flow ($\gamma = 1.4$) and compares the solution with the exact analytical values. The important conclusions are first, that the accuracy of the code is demonstrated by excellent agreement between computed and analytical solutions. That is, the maximum error for any variable at any CFL number is 3.0% with the average error generally below 0.5%.

Second, the characteristic boundary conditions discussed in Chapter 3 do satisfactorily "find" the correct set of inlet variables such that supercritical flow is achieved. Third, the computed Mach number at the throat deviates from unity by a maximum of 0.3%. Finally, the number of iterations decrease with increasing CFL (i.e. increasing Δt), but with an expected decrease in accuracy.

Numerical solutions for equilibrium calculations were validated by a comparison between computed results and; the law of mass action (10) for consistency, an equilibrium sonic speed at the throat, and for constant mass flow rate along the streamtube. In all cases where mass action was checked the agreement between the numerically computed concentration and an analytically computed concentration from equation (10) was accurate to six significant figures. The equilibrium Mach number departed from unity by at most 0.15% (for a CFL number equal to 10). As in the case of frozen flow, the accuracy of the numerical solution improved with an increase in mesh points or a decrease in CFL criteria. For all cases the steady state mass flow rate varied by at most 0.15% and for nonequilibrium flow the flux had the same accuracy as for isentropic flow at a given CFL number.

In addition to mass flow consistency, nonequilibrium flow was validated by verifying that the sonic point moved downstream of the minimum section as the rate coefficient (ϕ) increased (Equation 1). Also qualitative behavior was compared with the results of Reference [3]. For all cases, the equilibrium and frozen limit data banded that of nonequilibrium flow. For example, an increase in rate coefficient causes the species concentration to depart from near equilibrium farther downstream and the flow temperature to increase. It appears from the cited general results that the computer code does output results which are

consistent with the physics of the problem.

4.3 Integration Difficulties

Consider problems related to numerical instability as specifically due to the explicitly modeled source term. The species equation source term, \dot{w} , was evaluated at the current time level as outlined in Equation (50). Such a basis for estimation of the local change in concentration due to chemical reactions can be extremely inaccurate when the rate coefficient becomes large. Figure 6 illustrates the relative scale of the reaction rate compared to temporal step size. As the reaction coefficient increases, some flow regions tend to remain in near equilibrium, as is made clear by the slope of the fast reaction rate in Figure 6. Essentially, in such "fast" cases any small deviation from equilibrium produces a large initial rate to bring the concentration back to an equilibrium level. This initial rate quickly decreases as the chemical relaxation follows an exponential decay. If the integration step is a sufficiently small fraction of the relaxation time (τ_{chem}) such as used in Reference [10], the initial rate (or slope) will decrease at the next iteration level due to then updated values for dependent variables. However, if the time step exceeds the characteristic relaxation time, the concentration may (and most frequently does) overshoot the equilibrium level, thus leading to numerical instability.

For all nonequilibrium explicit cases, the basic predictor/corrector scheme was unstable unless the time step was taken as some fraction of the smallest relaxation time in the mesh. This constraint is apparent from the stability restriction of Equation (40).

When computing using the explicit/implicit algorithm, the local concentration can easily (and frequently does) overshoot the local current

equilibrium level without achieving numerical instability for rate coefficient magnitudes less than $\phi = 10^4$. A possible explanation is that the implicit stage of integration modifies the explicitly computed local evaluation by distributing these changes globally, thus effectively damping out oscillations. Nevertheless, as the rate coefficient increases such numerical damping is insufficient to maintain stability.

Four distinct methods were employed in attempting to improve numerical stability. Two methods used a cut off criteria to prevent the concentration from overshooting an equilibrium level. A third method used a point by point "type" splitting calculation for equilibrium and nonequilibrium regions. That is, either equilibrium or nonequilibrium equation were locally solved depending upon a type splitting criteria. A fourth technique consisted of a modification of an analytical/numerical approximation suggested by Bray [3]. A brief description of these algorithms will be presented in chronological order.

The "type splitting" (Method I) procedure employed a crossover criteria to determine which equation set (i.e. equilibrium or nonequilibrium) was to be solved at each mesh point (Figure 5). If the local reaction rate at a given grid point was sufficiently large such that the change in concentration exceeded the change to the current, local equilibrium level, identical equilibrium was assumed for that mesh point and time step. Therefore, equilibrium Equations (22) were applied at such points and nonequilibrium equations (19) at all other points in the mesh. This procedure met with some limited success. With this technique the explicit stability limit was, in fact, substantially increased. By eliminating the explicit relaxation time constraint, the temporal step limitation was determined by the CFL criteria. This algorithm proved to be

stable for reaction rates up to $\phi = 10^4$. To add an implicit stage (and hopefully further increase the system stability limit) required an assumption be made for changes in concentration ($\Delta\alpha$) at "equilibrium" points. For identical equilibrium, α is completely specified by the law of mass action. However, for this method any given mesh point in the computational domain could be in either equilibrium or nonequilibrium at different time levels. This required that a concentration change be specified regardless of the point type. Applying an estimated change did not improve stability. In fact, this method did not yield any stable implicit results.

A second method (II) employed an analytical approximation by Bray [3] to essentially remove the singular perturbation region. Bray assumed that in regions of the flow field for which the dissociation rate term was significantly larger than the species convection term, the flow was "in equilibrium." That is, the suggested criterion is:

$$(61) \quad \phi T^{1/2} \rho (1-\alpha) e^{-\Theta_d/T} \gg \left| u \frac{d\alpha}{dx} \right|$$

Bray considers a lower limit of 20 as the level at which nonequilibrium should be appropriate. A starting point was chosen based on this criterion. Figure 7 indicates that the procedure is not unstable, but does result in an oscillatory solution near the starting point. It was decided that the oscillations were related to the near equilibrium nature of the starting point, which lead to overshoot for large time steps.

The remaining two algorithms (III, IV) employed explicit damping of the species equation in order to improve and control stability. The first of these (Method III) used a cut off criterion based on the local, time dependent, equilibrium concentration. Any concentration change that

exceeded the change to the current local equilibrium level was restricted (cut off) so as to proceed at some fraction of the maximum possible change ($\alpha_1^* - \alpha_1$). This procedure produced numerically stable results for large time steps (CFL = 10) but did produce some "chatter" that could not be damped. The results of Figure 8 indicate the oscillations in a typical solution.

The final algorithm (Method IV) eliminated the chatter (of III) by basing a cut off criterion on the local steady state equilibrium concentration in contrast to the local, transient equilibrium. This procedure, in effect, provided a lower bound on the concentration distribution, and added the necessary stability to permit consideration of nonequilibrium flows with large time steps (Figure 9). The next section provides results that indicate the significant improvement that becomes available for nonequilibrium flow evaluation.

4.4 Results

Method IV algorithm implementation was as follows. First, the computation of both frozen and equilibrium limiting cases established an upper and lower bound to the nonequilibrium distribution behavior. The finite rate chemistry cases were then solved using an initial ($t = 0$) concentration distribution selected to be greater than the equilibrium lower bound over the entire domain. During the explicit portion of the algorithm, if an implied concentration change was larger in magnitude than that required to achieve local steady state equilibrium, the allowed change was "cut off" at some fraction of the difference. That is

$$\begin{aligned}
 &\text{if } \Delta \alpha > (\alpha_i^* - \alpha_i) \\
 (62) \quad &\text{then } \Delta \alpha = K(\alpha_i^* - \alpha_i) \\
 &\text{where } K \leq 1
 \end{aligned}$$

The entire domain was scanned at each step to locate the most downstream point for which a cross over of the equilibrium "stability" boundary would result if not constrained by the cut off condition (Equation (63)). The cut off rule was then applied at that point and all proceeding upstream points. By this means, an extremely inaccurate estimate of the concentration change was inhibited in the near equilibrium region. This effectively restricted the consequences of the local reaction rate so that the transient concentration distribution remained above that for the equilibrium limit. Essentially, imposing the lower bound provided the additional stability for the explicit/implicit algorithm in the singular perturbation region. That is, since the chemical reaction rate (\dot{w}), is time dependent and only evaluated at the current time level the inaccuracy of the explicit estimate of the rate is compensated by using a known lower bound to inhibit overshoot and oscillations. The individual solutions provide some evidence of the utility of Method IV.

The purpose of the completed sample cases was to explore the ability of the algorithm to produce results consistent with the physics, and to demonstrate the relative gain or penalty in computation time. Five cases (i.e., parameter sets) were considered in order to indicate a range of applicability and determine possible regions of difficulty. The five cases were as follows:

Case	α_0	CFL	ϕ
1	0.67	0.9	0,10 ³ ,5x10 ³ ,10 ⁴ 10 ⁵ , ∞
2	0.67	5.0	0,10 ³ ,5x10 ³ ,10 ⁴ ,10 ⁵ , ∞
3	0.67	10.0	0,10 ³ ,5x10 ³ ,10 ⁴ , ∞
4	0.40	0.9	0,10 ⁴ , ∞
5	0.40	5.0	0,10 ⁴ , ∞

In all cases, a segmented solution procedure was employed using 57 points between computational boundaries. The first segment, contained the minimum section which expanded the fluid to supersonic velocity ($Y_{\max} = 2.0$). The second, completing supersonic section expanded the flow further over $2.0 < y < 10.0$. The large area ratio was introduced so as to demonstrate chemical freezing even more clearly. The convergence criterion (Equation (60)) was $\epsilon = 10^{-4}$. The initial conditions for all computed solutions were the same. That is, for the first segment ($Y_{\max} = 2.0$) a decreasing linear distribution for density, velocity, and temperature and a constant concentration distribution. When comparing iterations to convergence for a given stagnation condition, the same initial profile was always employed. Table 3-7 summarize iterations and cut off criteria for each case. When comparing relative near time note that an explicit/implicit iteration is 2.25 times an explicit iteration.

4.4.1 Case 1 - Hot Gas Explicit Solution (CFL = 0.9)

Results for explicit integration (CFL = .9) are illustrated in Figure 10-14 and iterations and cut off conditions are summarized in Table 3. The concentration profiles of Figure 11 demonstrate the freezing phenomenon as supersonic flow continues to expand. In addition, this figure shows the tendency of the concentration to maintain a near equilibrium distribution for larger area ratios for increasing ϕ . The initial spatial departure from equilibrium is more easily seen in Figure 11 which is an exploded view of the throat region. The temperature, velocity, and density follow expected trends for nonequilibrium flow. That is, temperature and velocity increase and density decreases with increasing ϕ . An interesting consequence of applying the cut off algorithm is that the explicit integrate stage (generally restricted by chemistry dominated flows) may

then be integrated using only a CFL stability criterion. More will be said regarding the quantitative improvements after the implicit results are discussed.

4.4.2 Case 2 - Hot Gas Explicit/Implicit Solution (CFL = 5.0)

Some numerical results obtained for this explicit/implicit case are summarized in Table 4. The graphed distributions (Figures 15-19) of dependent variables (ρ, u, T, α) indicates excellent agreement with the explicit (CFL = 0.9) solution. For example the exit concentrations for both $\phi = 10^4, 10^5$ differ by less than 0.1% for either ϕ . Also, from a comparison of Figures 11 and 16 it can be seen that the concentrations at the intermediate $y = 2.0$ are indeed virtually identical. However, the implicit (CFL = 5.0) algorithm converged in approximately 0.56 the time required for the explicit (CFL = 0.9) method.

4.4.3 Case 3 - Hot Gas Explicit/Implicit Solution (CFL = 10.0)

When the CFL number was increased to 10.0 a significant decrease in accuracy was noted for higher rate constants. Table 5 lists the iteration and cut off coefficient (K) for the results plotted in Figures 20-24. Note that results for $\phi > 10^4$ are not presented for CFL = 10.0. In order to obtain a sufficiently accurate solution for $\phi = 10^4$ a cut off criterion of 0.2 was necessary. This increased the number of iterations to 300, three times the number for CFL = 5.0 (compare Tables 4 and 5). Figure 25 illustrates the substantial inaccuracy that corresponds to a $\phi = 10^4$ evaluation with $K = 1.0$ used as a cut off basis. The plot of those results (inverted triangles in Figure 25) indicates a virtual equilibrium concentration distribution out to $y = 2.0$. However, both Cases 1 and 2 indicated that departure from equilibrium occurred at a substantially smaller area ratio for the same rate parameter (see Figures 10 and 15). A

decrease to $K = 0.2$ (larger values were tried) led to solutions for Cases 2 and 3 which were nearly identical, as indicated by the upper concentration distribution in Figure 25, but with a run time increased by a factor of three.

A measure of the error for $\phi = 10^4$ and the range of CFL numbers is plotted in Figure 26. The error is defined as:

$$(63) \quad \text{error} = \frac{\left[\sum_{i=1}^I (U^{n+10} - U^n)^2 / I \right]^{1/2}}{\left[\sum_{i=1}^I (U^{10} - U^1)^2 / I \right]^{1/2}}$$

The CFL = 5 case converges much more rapidly than the CFL = 0.9 case (100 uses 410 iterations) and produces virtually identical results. Similarly, for CFL = 10 with a $K = 1.0$ cut off condition, convergence is quicker than for either 0.9 or 5.0, but as discussed above (Figure 25) the results are quite inaccurate. The $K = 0.2$ iteration history plot (Figure 26) indicates the substantial increase in iteration count (from 60 to 300) and suggests that CFL = 10.0 would be less efficient than an explicit case for $\phi = 10^5$. That is, when the acceptable error level is below that corresponding to the explicit stability limit then an explicit method becomes preferable to an implicit method (Figure 27). The conclusion is that for a sufficiently large CFL number the inaccuracy introduced by the implicit method produces a severe constraint on the cut off condition. Therefore, at sufficiently high ϕ , a lower CFL number is favorable for a more accurate solution in fewer iterations.

4.4.4 Comparison of Computation Time

The improvement in number of iterations and computer time can be seen from Figures 28 and 29. Results for Cases 1 and 2 are compared with an explicit calculation employing τ_{chem} stability constraint instead of the

cut off algorithm. As discussed, for reaction dominated flows the explicit time step is determined by the characteristic relaxation time. (e.g. for an upwind algorithm equation (40) determines the step size). In practice, numerical investigation has indicated [10] that for stability in a reaction dominated flow

$$(64) \quad \Delta t = \beta (\tau_{chem})_{min}$$

where

$$\beta < 1$$

That is, for stability the integration must proceed at some fraction of the smallest relaxation time in the field. The results shown in Figure 28 for the explicit integration without the cut off criterion were computed with $\beta = 0.3$. This value was determined by decreasing β until a stable solution was obtained. The results plotted in Figure 28 indicate that the " τ_{chem} " case deviates from the others downstream of the throat. As would be expected the cut off algorithm does introduce solution error but the results differ by only 0.1% at $y = 2.0$. The other dependent variable (ρ, u, T) indicate similar results with the maximum error at any mesh point being less than 3.0%. The significant improvement in computer time realized for this small decrease in accuracy is summarized below:

<u>Method</u>	<u>Iterations</u>	<u>Average Δt</u>	<u>Relative Run Time</u>
Explicit - CFL = .9 $\beta = 0.3$	3470	0.04	1.0
Explicit - CFL = .9 $K = 1.0$	430	0.2	0.15
Implicit - CFL = 5.0 $K = 0.5$	100	1.0	0.08

This chart illustrates a drastic improvement in run time for a small decrease in accuracy. The iteration history plot (Figure 29) shows graphically the difference in convergence rate.

4.4.5 Cases 4 and 5 - Cold Gas

For completeness, results for two CFL numbers with lower stagnation conditions (corresponding to $\alpha_0 = 0.40$) are illustrated in Figures 30-33 for CFL = 0.9 and Figures 34-37 for CFL = 5.0 and iteration and cut off conditions are summarized in Tables 6 and 7. The graphs and tables indicate quite similar qualitative behavior and conclusions with regard to the numerical algorithm consideration.

4.4.6 Boundary Conditions

Characteristic boundary conditions were presented in Chapter 2 and the numerical procedure was discussed in Chapter 3. The characteristic boundary condition provided stable inlet condition without overspecifying the problem, which otherwise may have prevented the automatic development of a proper supercritical flow. As discussed in the section on code validation (Chapter 4), a choked flow condition was achieved using this characteristic treatment of boundaries. Figures 38 and 39 indicate the solution of the inlet velocity to a steady state value consistent with critical mass flow and unit Mach number at the minimum section for an equilibrium ($\phi = \infty$) calculation. Figure 39 is an enlargement of Figure 38 for iteration count less than 100. For all CFL numbers the inlet velocity overshoots the steady state value with subsequent oscillation to a final value.

Chapter 5

CONCLUSIONS AND RECOMMENDATIONS

Considerable savings in computing time has been demonstrated using an explicit/implicit algorithm in conjunction with a cut off condition on the explicitly computed reaction rate (\dot{w}). For results presented in this paper, the explicit/implicit algorithm was two to three times faster than the explicit stage alone, when the cut off condition was employed for both the explicit and explicit/implicit methods. In addition, when comparing this algorithm with other explicit integration methods for stiff equations such as presented in Reference [10] the explicit stage was seven times and the explicit/implicit algorithm twelve times faster for the particular case investigated ($\phi = 10^5$). MacCormack's explicit/implicit method [15] was proven to work for a system of stiff equation and with the addition of the cut off algorithm was extended to cover a wider range of nonequilibrium flow ($0 < \phi < 10^5$).

Characteristic boundary conditions have also been devised for quasi-one dimensional supercritical flow. These boundary conditions permitted the solution of an unsteady system of equation (to a steady state final value) without a priori specification of inlet conditions consistent with supercritical mass flow. Results presented indicate that with these characteristic boundary conditions the Mach number at the minimum sections for the limiting equilibrium and frozen cases was accurate to 0.15%.

An extension of the approach may be fruitful in three areas: 1) increased number of dimension; 2) higher order reaction systems; 3) different integration algorithms. The extension of the current method to higher dimensions will require careful consideration of the stability

boundary concept when applying a cut off condition. It may be possible to simply apply the cut off condition in an analogous manner. That is, solve the two dimension equilibrium case and use this as the local lower bound. For higher order reaction systems possible concentration stability boundaries might be defined by considering the equilibrium distribution of each species with all other species frozen. Finally, for improved integration techniques, a more extensive parametric study of cut off condition with reaction rate coefficient and CFL number must be performed to provide more data regarding the optimum cut off criterion combination in terms of a given reaction rate coefficient and step size. In addition, as discussed in Chapter 3, inclusion of the source term in the implicit stage of integration should be investigated to determine if numerical instabilities due to explicit evaluation of the rate term can be eliminated without losing the advantage of the bidiagonal structure of the explicit/implicit algorithm.

In all, substantial improvements in computation time for a system of stiff equations have been demonstrated. Application of characteristic boundary conditions in conjunction with an unsteady system of Euler equations has eliminated the two point boundary value problem. The cut off algorithm has extended the range of applicability of the MacCormack explicit/implicit scheme. The success of this numerical method provides a basis for extension to higher dimensions and more complex reacting systems.

REFERENCES

- [1] Liepmann, H.W., Roshko, A., Elements of Gasdynamics. John Wiley, New York, 1957.
- [2] Hall, J.G., Treanor, C.E., "Nonequilibrium Effects in Supersonic-Nozzle Flows." AGARDograph 124, December 1967.
- [3] Bray, K.N.C., "Chemical and Vibrational Nonequilibrium in Nozzle Flows." Nonequilibrium Flows Part II, ed. Peter P. Wegner, Marcel Dekker, Inc., New York, 1970.
- [4] Hall, J.G., Russo, A.L., "Studies of Chemical Nonequilibrium in Hypersonic Nozzle Flows." Cornell Aeronautical Laboratory, Report AD-1118-A-6, AFOSR TN 59-1090, 1959.
- [5] Lomax, H., Bailey, H., "A Critical Analysis of Various Numerical Integration Methods for Computing the Flow of a Gas in Chemical Nonequilibrium." NASA TND-4109, August 1967.
- [6] Eschenroeder, A.Q., Boyer, D.W., Hall, J.G., "Nonequilibrium Expansions of Air with Coupled Chemical Reactions." The Physics of Fluids, Volume 5, Number 5, May 1962.
- [7] Treanor, C.E., "A Method for the Numerical Integration of Coupled First-Order Differential Equations with Greatly Different Time Constants." Mathematics of Computation, Volume 20, 1966.
- [8] Kee, R.J., Dwyer, H.A., "Review of Stiffness and Implicit Finite-Difference Methods in Combustion Modeling." Progress in Astronautics and Aeronautics Volume 76, eds. J. Ray Bowen et al, 1979.
- [9] Serra, R.A., "The Determination of Internal Gas Flows by a Transient Numerical Technique." AIAA Paper 71-45, 1971.
- [10] Anderson, J.D., "A Time-Dependent Analysis for Vibrational and Chemical Nonequilibrium Nozzle Flows." AIAA Journal, Volume 8, Number 3, March 1970.
- [11] Lax, P.D., Wendroff, B., "Difference Schemes for Hyperbolic Equations with High Order of Accuracy." Communications on Pure and Applied Mathematics. Volume 17, 1964.
- [12] McCormack, R.W., "The Effect of Viscosity in Hypervelocity Impact Cratering." AIAA Paper 69-354, May 1969.
- [13] Vincenti, W.G., Kruger, C.H., Introduction to Physical Gas Dynamics. John Wiley, New York, 1965.
- [14] Lighthill, M.J., "Dynamics of a Dissociating Gas Part I: Equilibrium Flow." Journal of Fluid Mechanics, Volume 2, 1957.

- [15] Clarke, J.F., "The Linearized Flow of a Dissociating Gas." Journal of Fluid Dynamics. Volume 7, 1960.
- [16] Roache, P.J., Computational Fluid Dynamics. Hermosa Publishers, Albuquerque, 1976.
- [17] MacCormack, R.W., "A Numerical Method for Solving the Equations of Compressible Viscous Flow." AIAA Journal, Volume 20, Number 9, September 1982.
- [18] Kutler, P., Lomax, H., "Shock-Capturing, Finite-Difference Approach to Supersonic Flows." AIAA Paper 71-99, January 1971.
- [19] Beam, R.M., Warming, R.F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations." AIAA Third Computational Fluid Dynamics Conference, Albuquerque, New Mexico, 1977.
- [20] Moretti, G., "Importance of Boundary Conditions in the Numerical Treatment of Hyperbolic Equations." The Physics of Fluids, supplement 11, 1969.
- [21] Gustafsson, G., Oliger, J., "Stable Boundary Approximations for a Class of Time Discretizations of $u_t = A D u$." OPPSALA University Department of Computer Science, Report Number 87, September 1980.
- [22] Chakravarthy, S.R., "Euler Equations -Implicit Schemes and Implicit Boundary Conditions." AIAA Paper 82-0228, January 1982.
- [23] Ames Research Staff, "Equations, Tables, and Charts for Compressible Flow." NACA Report 1135, 1953.
- [24] Warming, R.F., Beam, R.M., Hyett, B.J., "Diagonalization and Simultaneous Symmetrization of the Gas-Dynamic Matrices." Mathematics of Computation, Volume 29, Number 132, October 1975.

Table 1

INVESTIGATED NUMERICAL MODEL PARAMETERS

<u>Method</u>	<u>CFL Number</u>
Explicit	0.9
Explicit/Implicit	5.0, 10.0
 <u>Initial Conditions</u>	
Dependent Variables (ρ, u, T, α)	<ul style="list-style-type: none"> - Constant stagnation level - Linear between spatial boundaries - Equilibrium distribution
 <u>Configuration</u>	
Maximum area ratio for parabolic distribution	<u>Y_{max}</u> 2.0351, 10.31
 <u>Stagnation Conditions</u>	
Concentration	<u>α_0</u> 0.4, 0.67

Table 2

MACCORMACK EXPLICIT/IMPLICIT
ALGORITHM WITH CHARACTERISTIC BOUNDARY
CONDITIONS

Input data: $\gamma = 1.4$, $Y_{\max} = 2.0351$,
No. mesh points = 57, $\epsilon = 10^{-6}$

<u>Location</u>	<u>CFL</u>	<u>ρ</u>	<u>u</u>	<u>T</u>	<u>Max % Error</u>	<u>Iterations</u>
Inlet	Exact	0.9564	0.2973	0.9823	-	-
	0.9	0.9563	0.2976	0.9823	0.1	600
	5.0	0.9565	0.2975	0.9824	0.07	120
	10.0	0.9574	0.2977	0.9827	0.13	80
Throat	Exact	0.6339	0.9128	0.8333	-	-
	0.9	0.6341	0.9129	0.8333	0.03	600
	5.0	0.6337	0.9114	0.8330	0.16	120
	10.0	0.6338	0.9091	0.8327	0.40	80
Outlet	Exact	0.1806	1.5743	0.5044	-	-
	0.9	0.1809	1.5741	0.5051	0.17	600
	5.0	0.1817	1.5730	0.5081	0.73	120
	10.0	0.1827	1.5721	0.5119	3.0	80

Table 3

CASE 1 SUMMARY
 $\alpha_0 = 0.67$, CFL = 0.9, $\epsilon = 10^{-4}$

ϕ	<u>2</u>		<u>Y_{max}</u>	
	Iterations	K	Iterations	K
0	400	-	180	-
10^3	410	1.0	180	-
5×10^3	400	1.0	190	-
10^4	410	1.0	200	-
10^5	430	1.0	220	1.0
∞	460	-	210	-

Table 4

CASE 2 SUMMARY
 $\alpha_0 = 0.67$, CFL = 5.0, $\epsilon = 10^{-4}$

ϕ	<u>2</u>		<u>Y_{max}</u>	
	Iterations	K	Iterations	K
0	90	-	60	-
10^3	100	1.0	60	-
5×10^3	90	1.0	60	-
10^4	100	1.0	60	-
10^5	100	0.5	70	1.0
∞	100	-	60	-

Table 5

CASE 3 SUMMARY
 $\alpha_0 = 0.67$, CFL = 10.0, $\epsilon = 10^{-4}$

ϕ	<u>2</u>		<u>10</u>	
	Iterations	K	Iterations	K
0	80	-	50	-
10^3	60	1.0	50	-
5×10^3	60	1.0	50	-
10^4	60	1.0	60	1.0
	300	0.2	60	0.5
"	60	-	50	-

Table 6

CASE 4 SUMMARY
 $\alpha_0 = 0.4$, CFL = 0.9, $\epsilon = 10^{-4}$

ϕ	<u>2</u>		<u>10</u>	
	Iterations	K	Iterations	K
0	410	-	180	-
10^4	430	1.0	200	-
"	460	-	210	-

Table 7

CASE 5 SUMMARY
 $\alpha_0 = 0.4$, CFL = 5.0, $\epsilon = 10^{-4}$

<u>ϕ</u>	<u>Y_{\max}</u>			
	<u>2</u>		<u>10</u>	
	Iterations	K	Iterations	K
0	90	-	60	-
10^4	100	1.0	60	-
∞	100	-	70	-

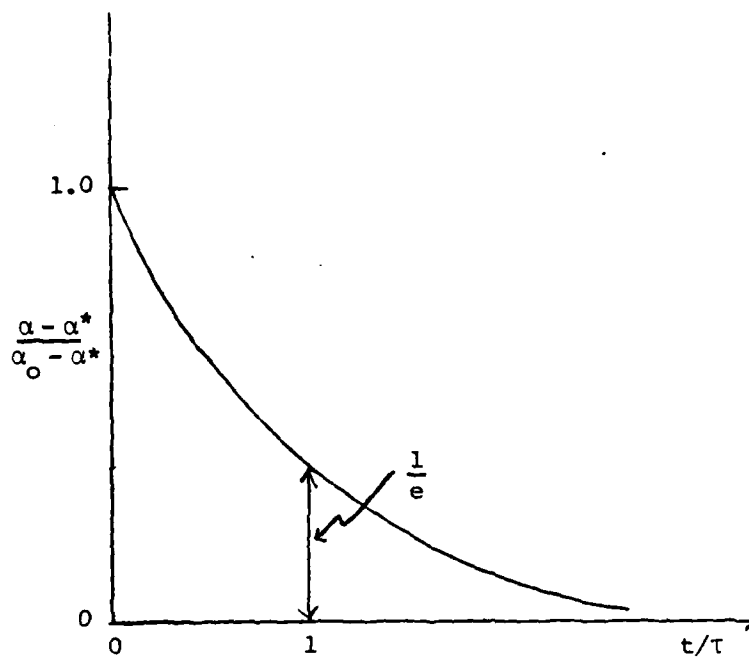


Fig. 1. Illustration of Relaxation Time

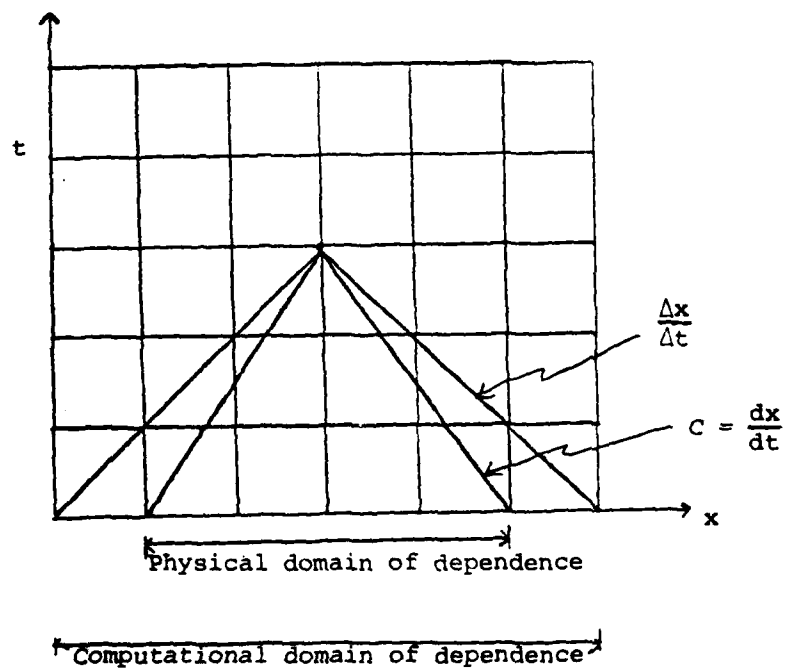


Fig. 2. Domain of Dependence for Hyperbolic Equations

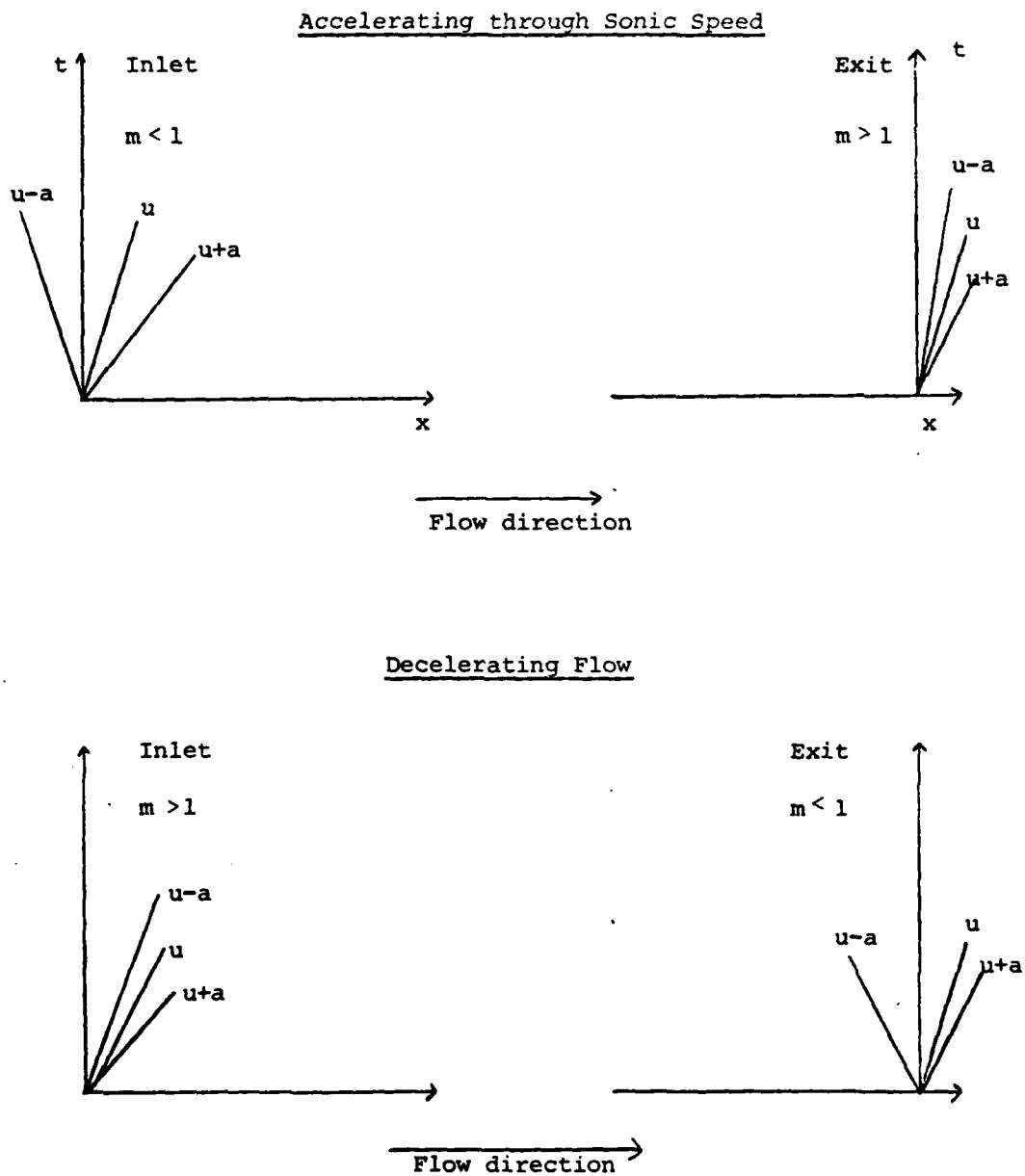


Fig. 3. Characteristic Boundary Conditions

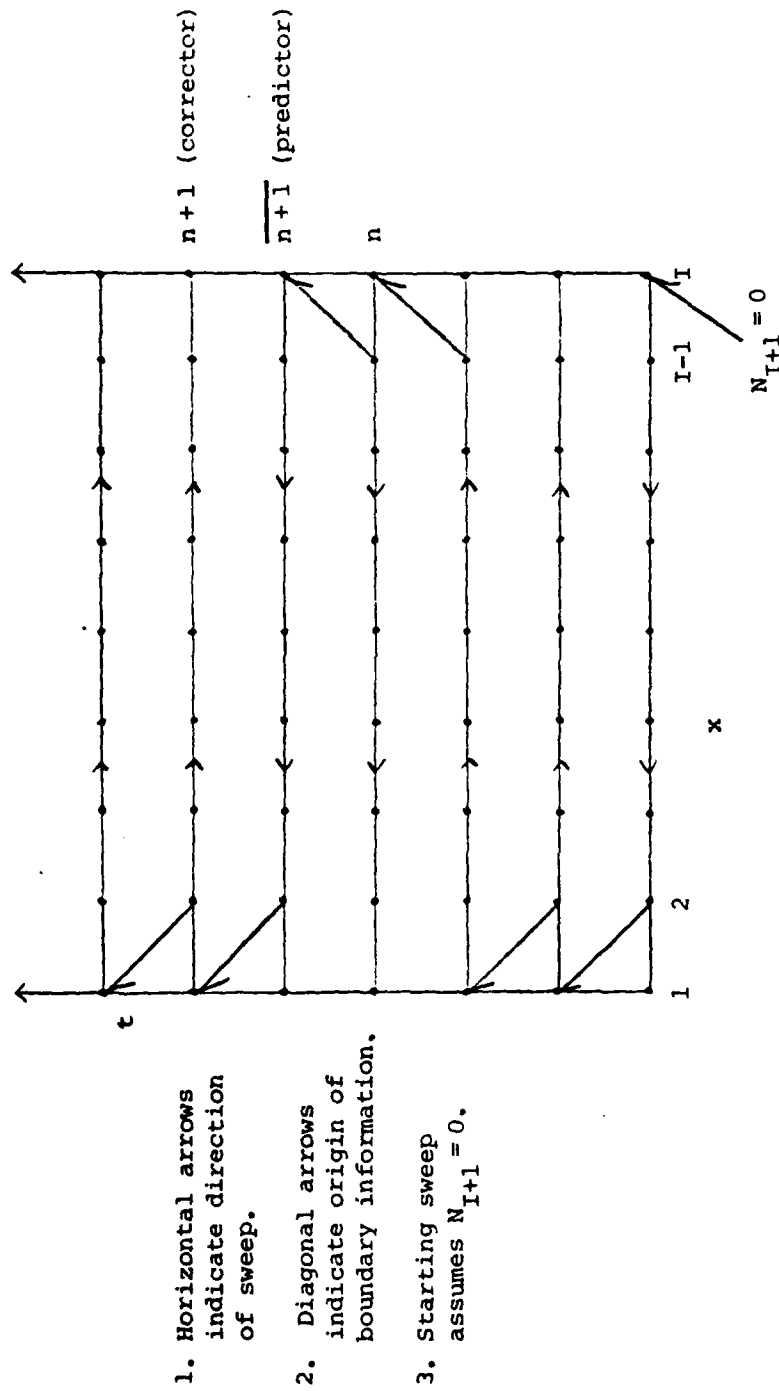


Fig. 4. Bidirectional Matrix Sweep with Boundary Conditions

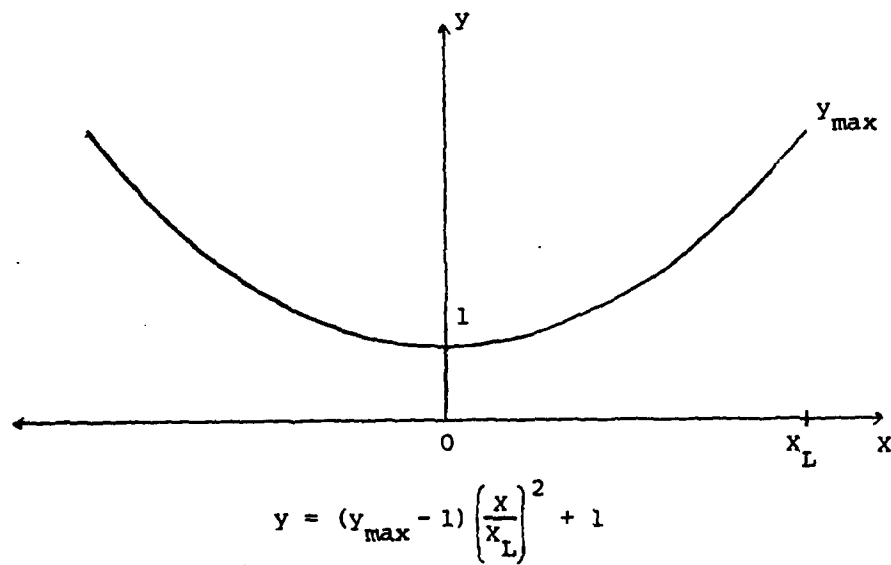
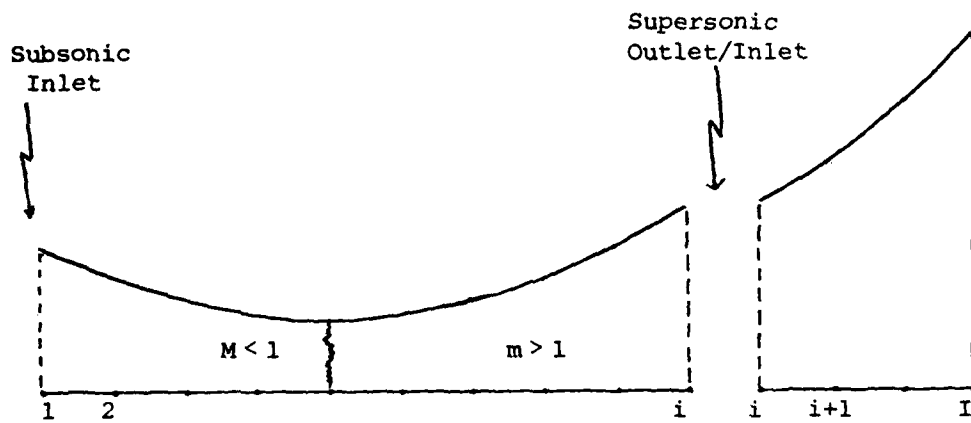


Fig. 5. Streamtube Segmentation and Parabolic Area Distribution

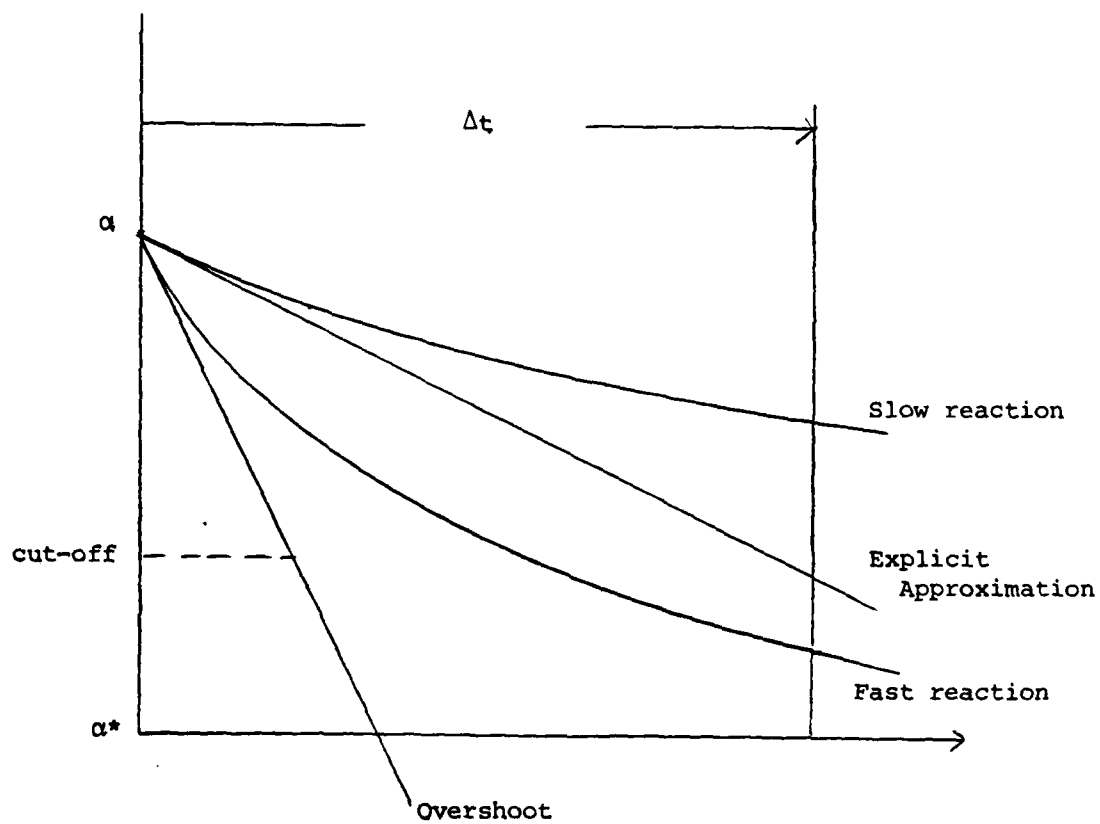


Fig. 6. Overshoot and Cut-off Condition

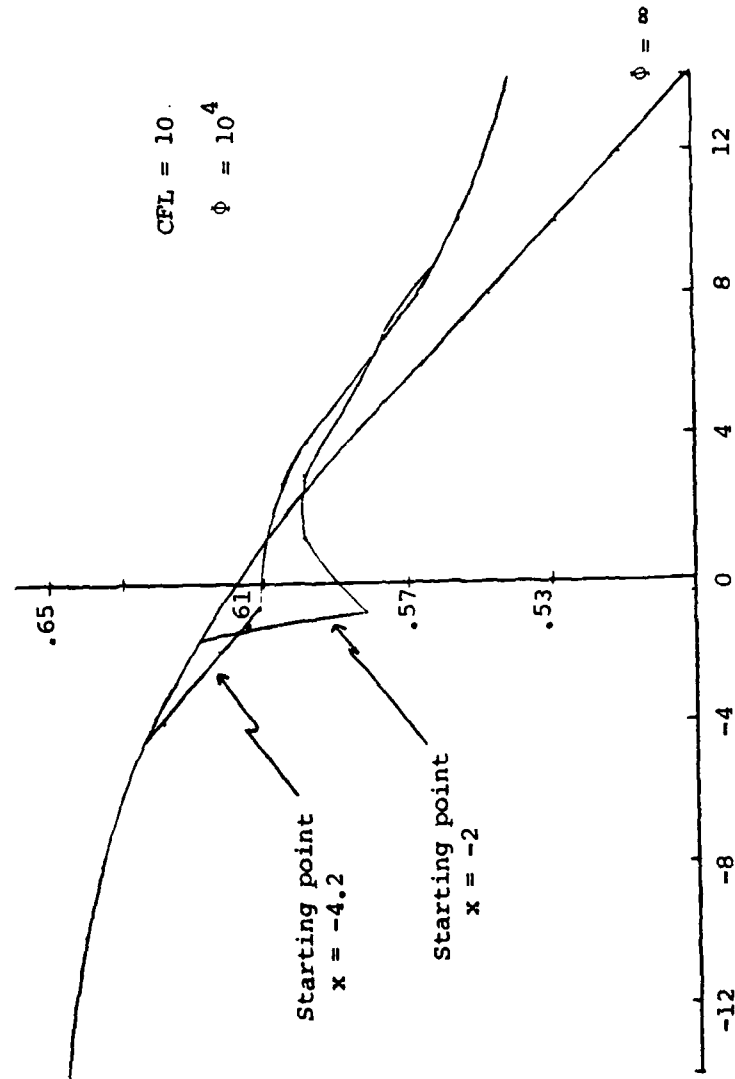


Fig. 7. Sample Result from Application of Bray Criterion

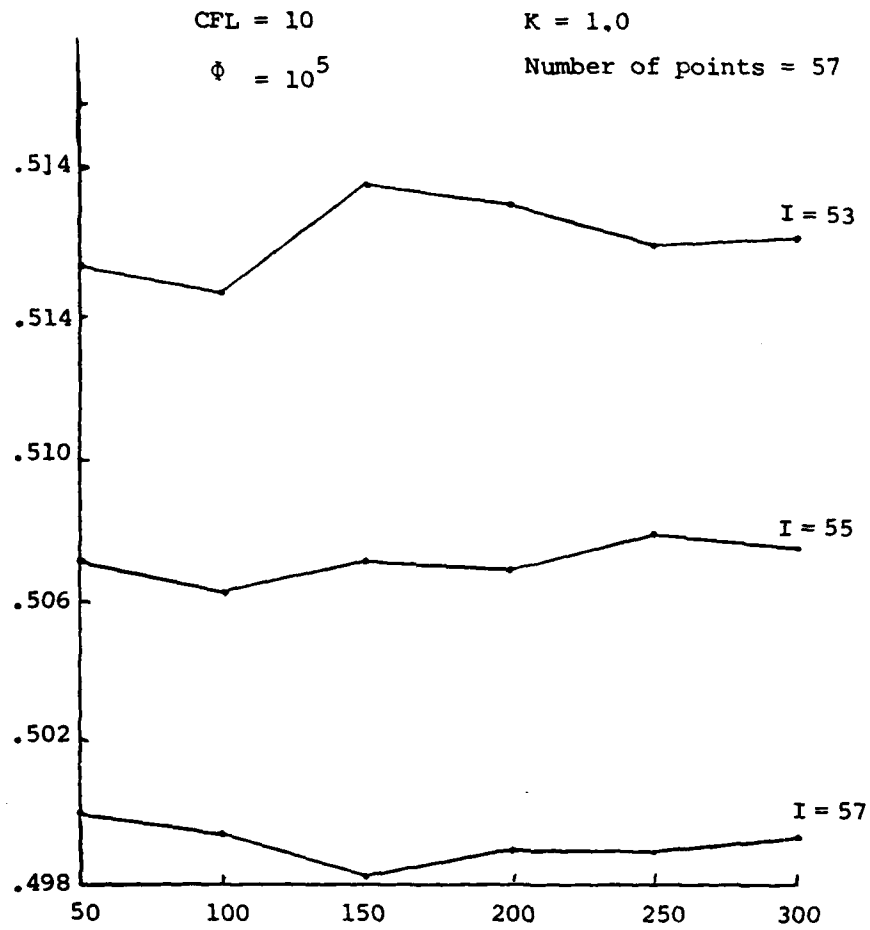


Fig. 8. Oscillations in Transient Equilibrium
Cut-Off Algorithm

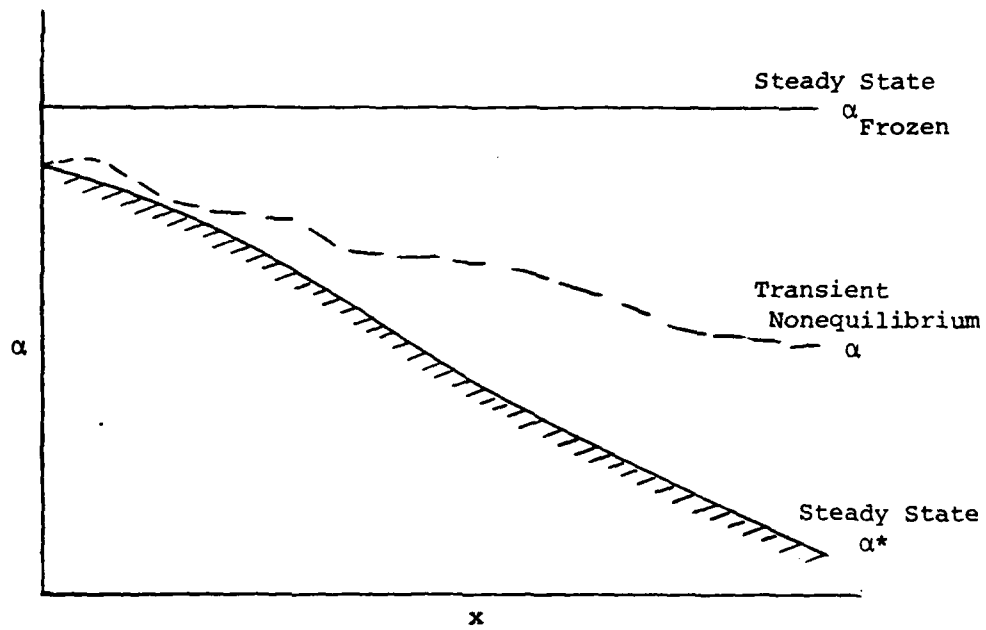
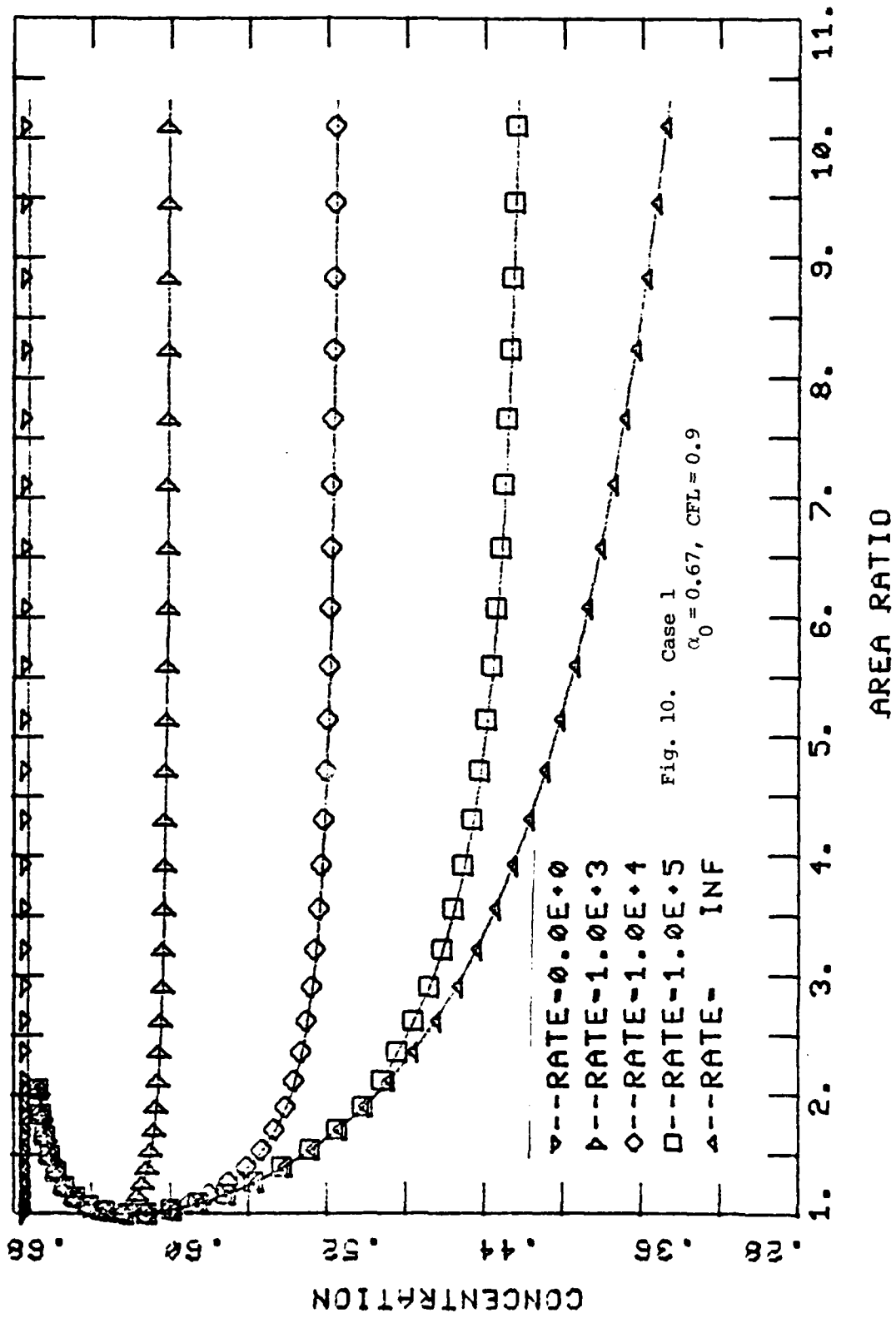
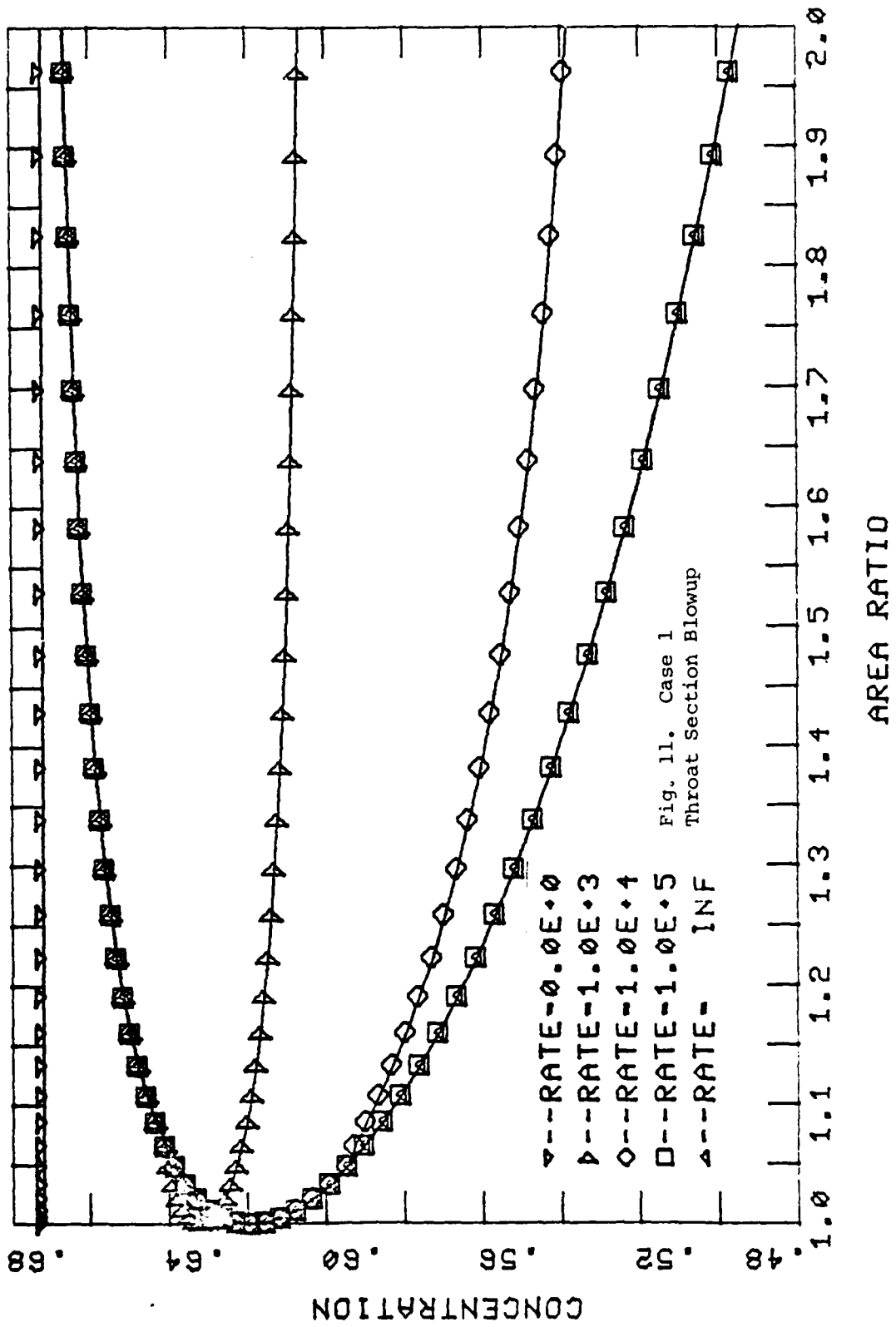


Fig. 9. Steady State Equilibrium Concentration Distribution Boundary

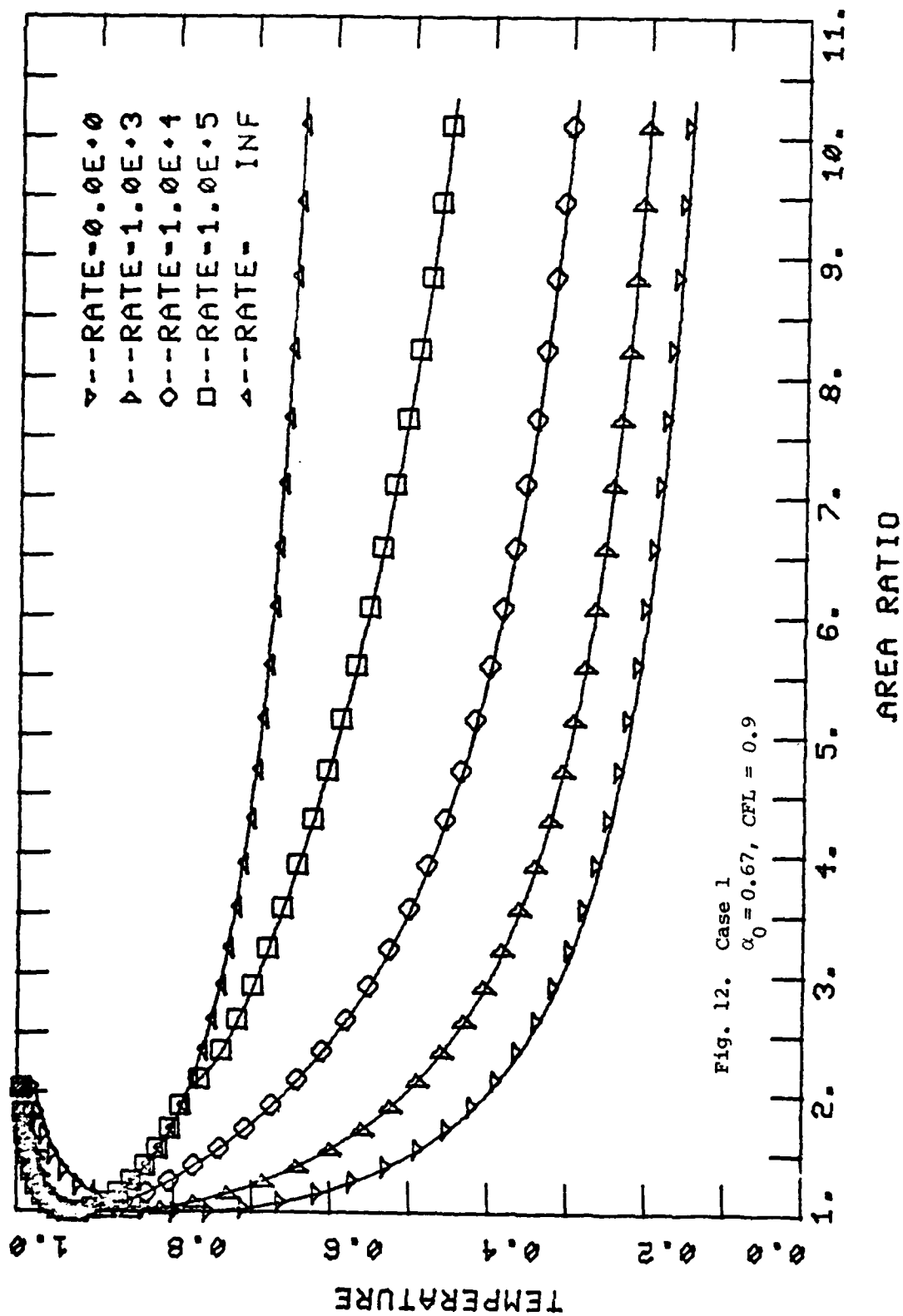
CONCENTRATION PROFILES



CONCENTRATION PROFILES



TEMPERATURE PROFILES



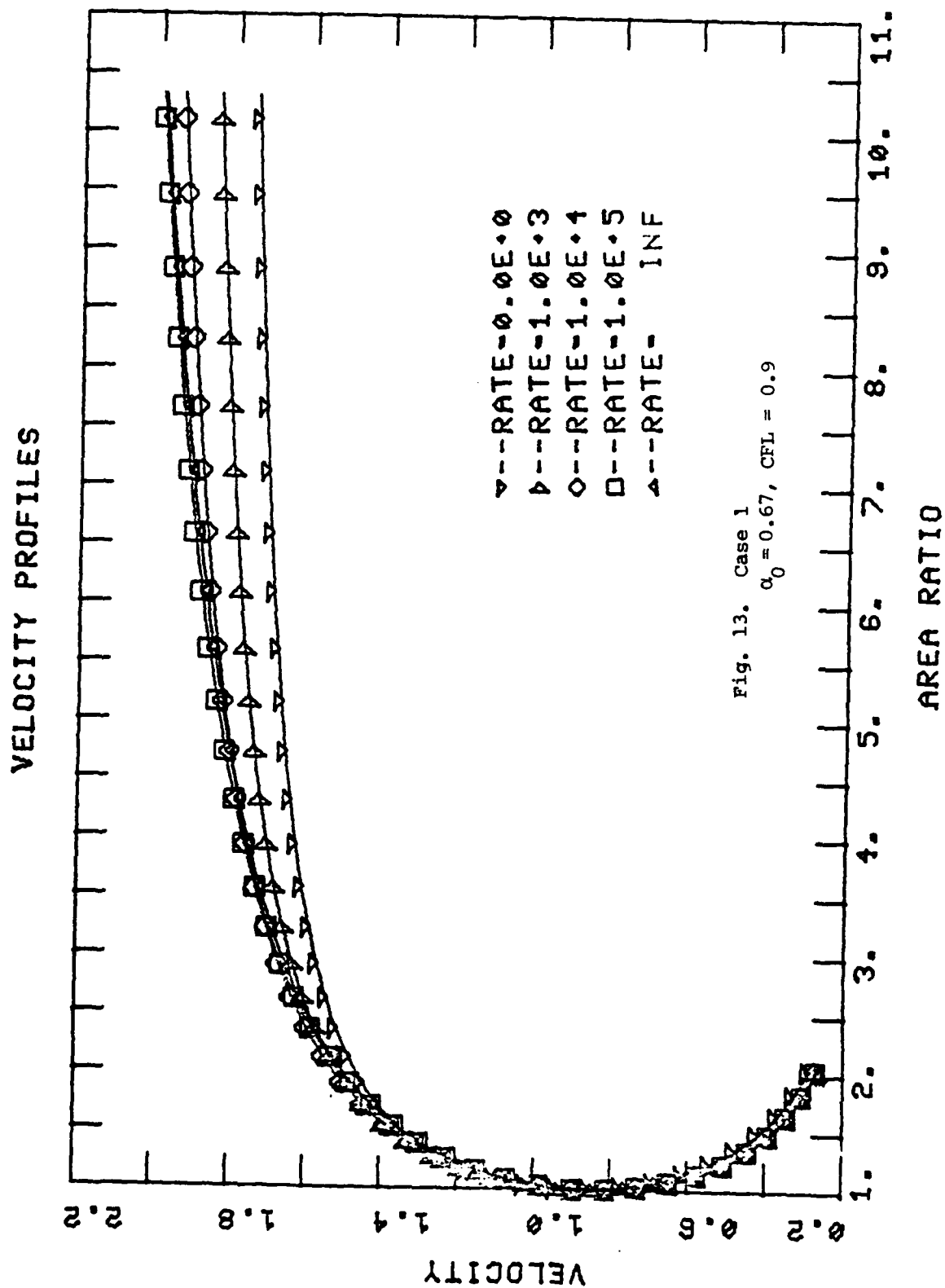
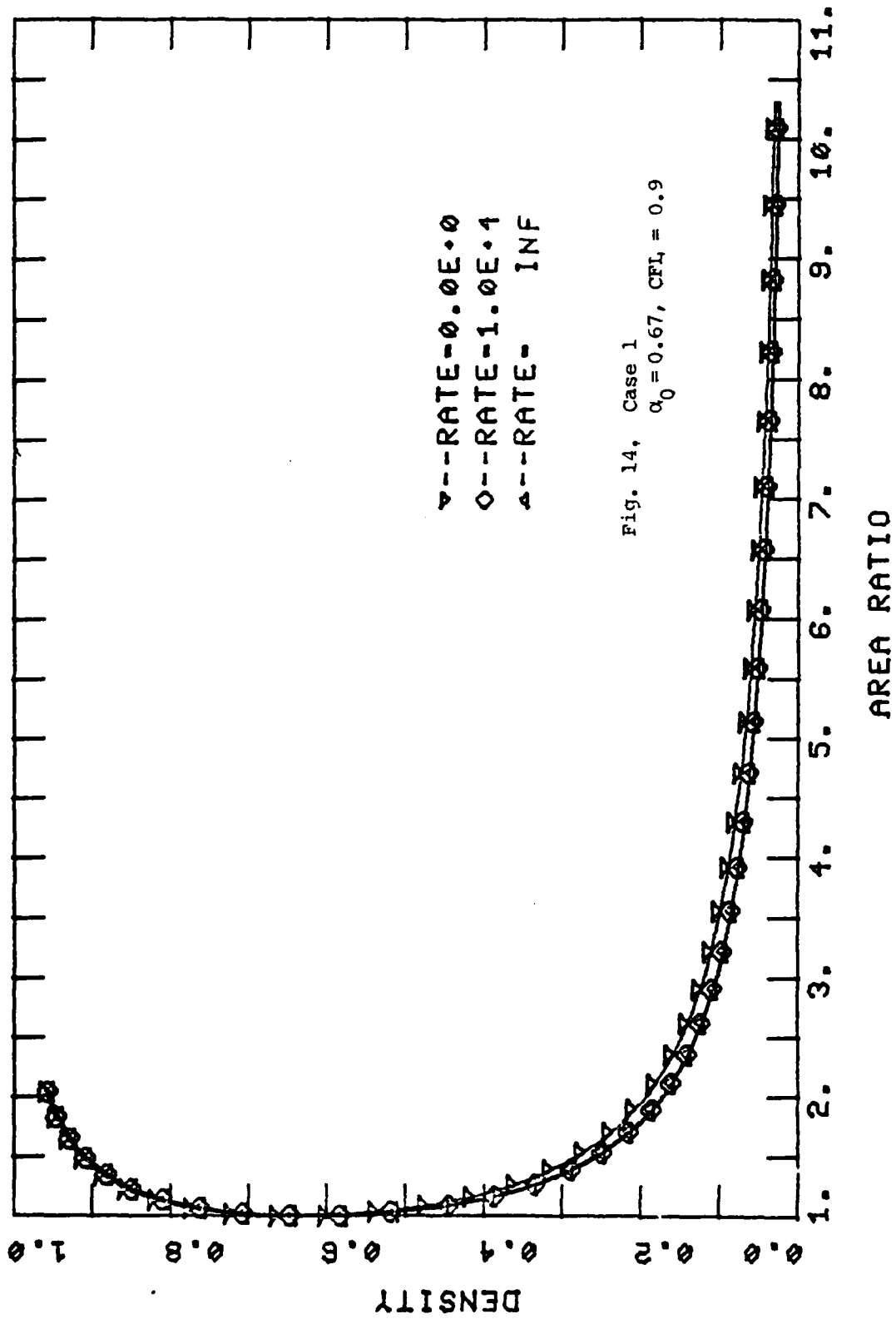
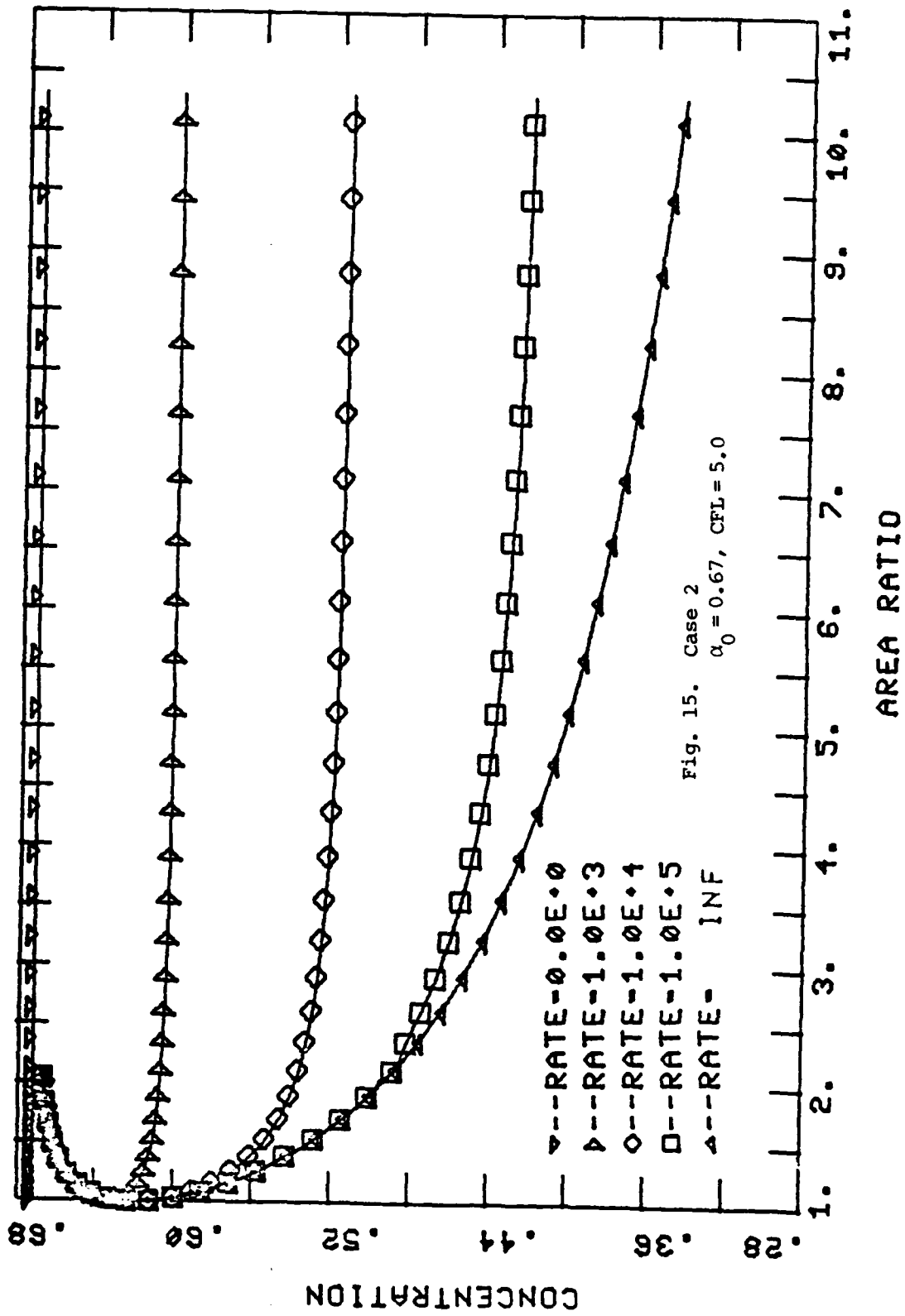


Fig. 13. Case 1
 $\alpha_0 = 0.67$, CFL = 0.9

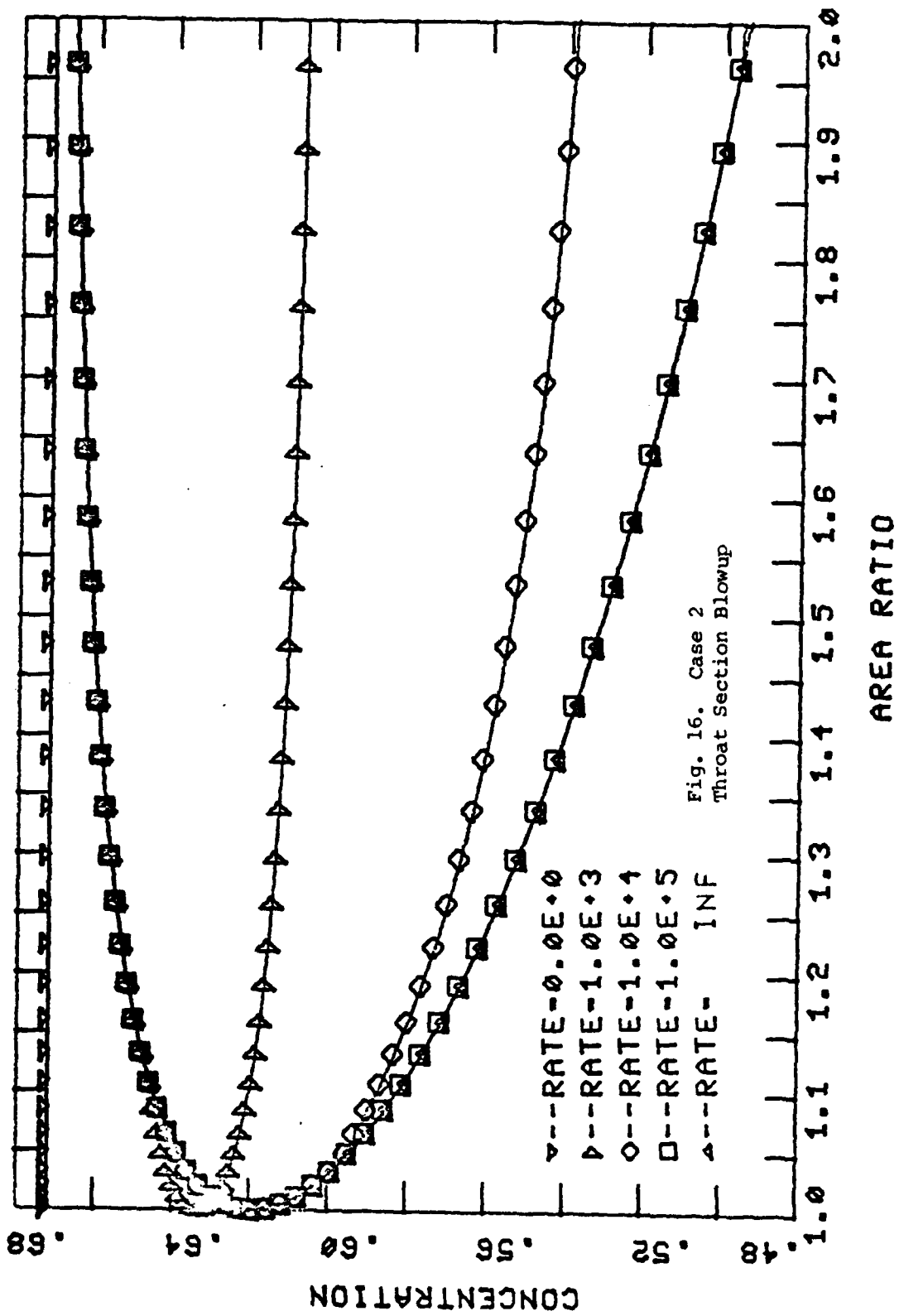
DENSITY PROFILES



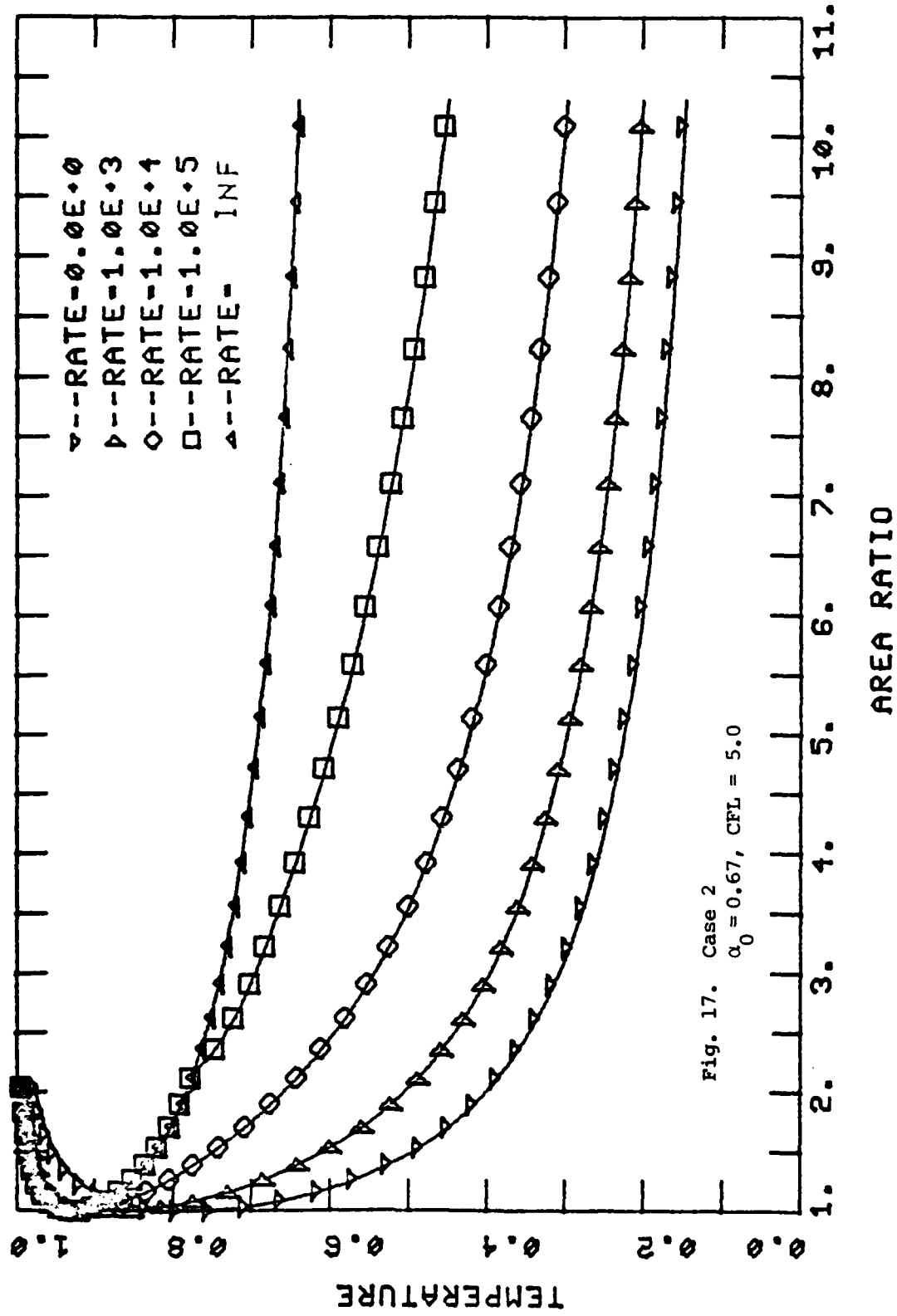
CONCENTRATION PROFILES



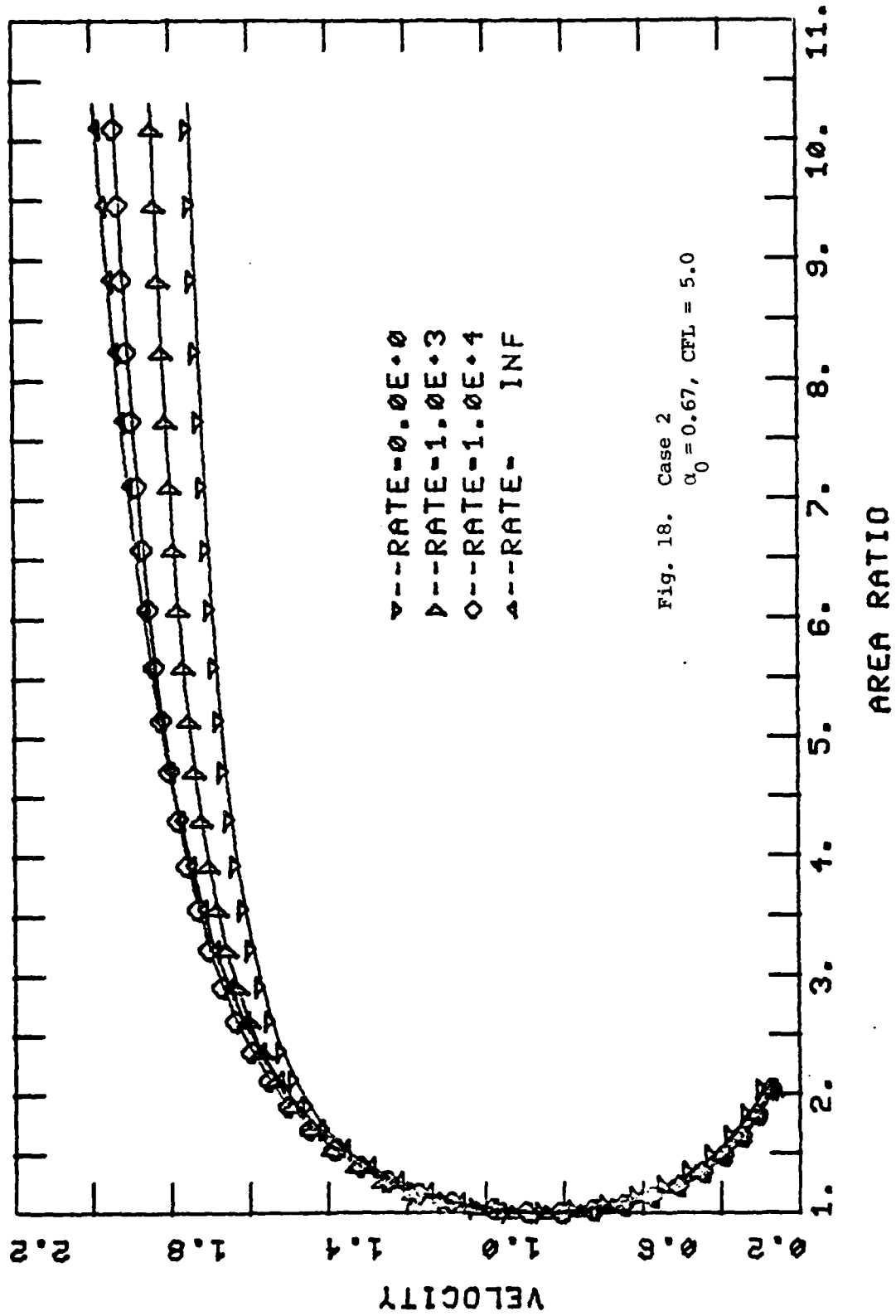
CONCENTRATION PROFILES



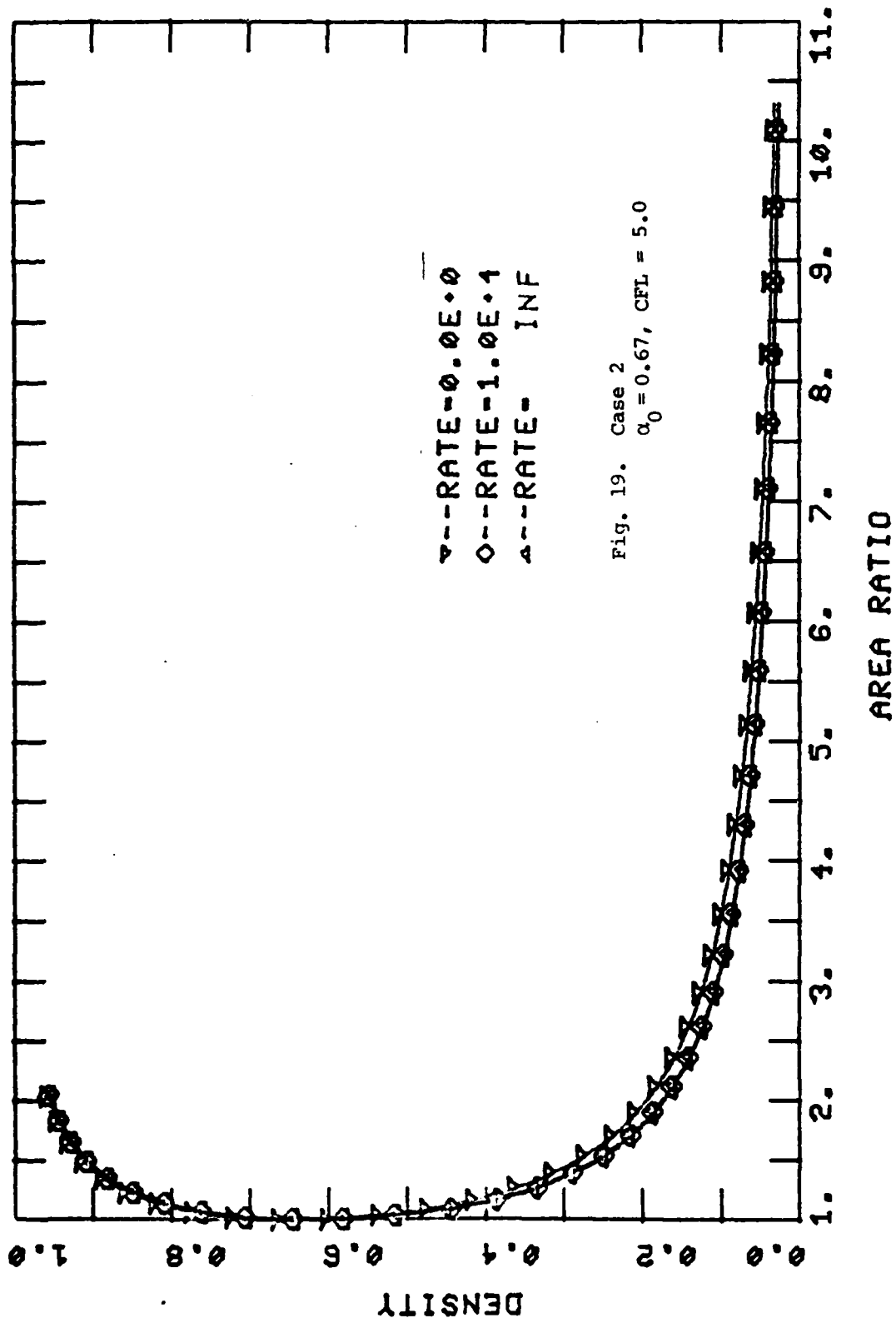
TEMPERATURE PROFILES



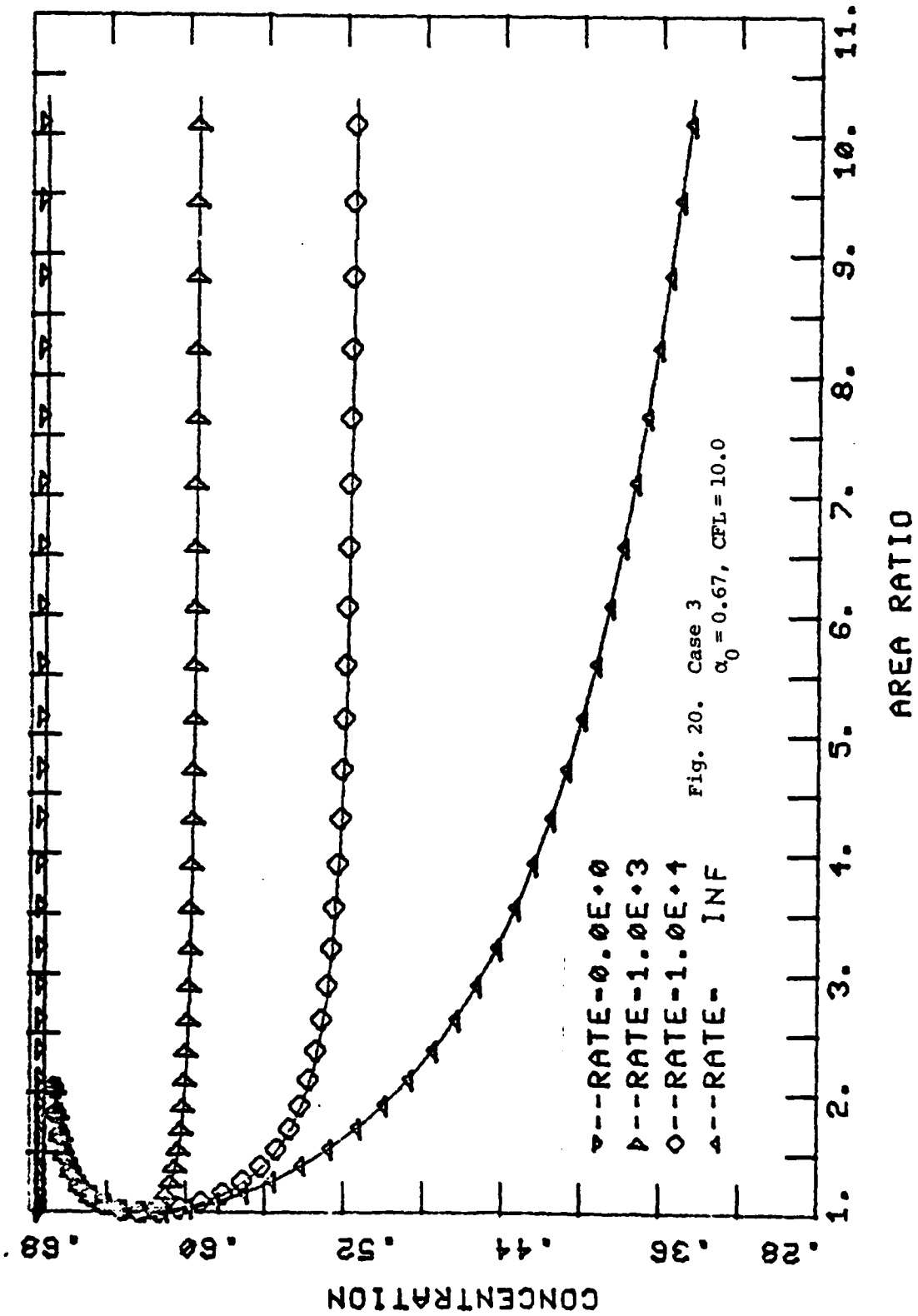
VELOCITY PROFILES

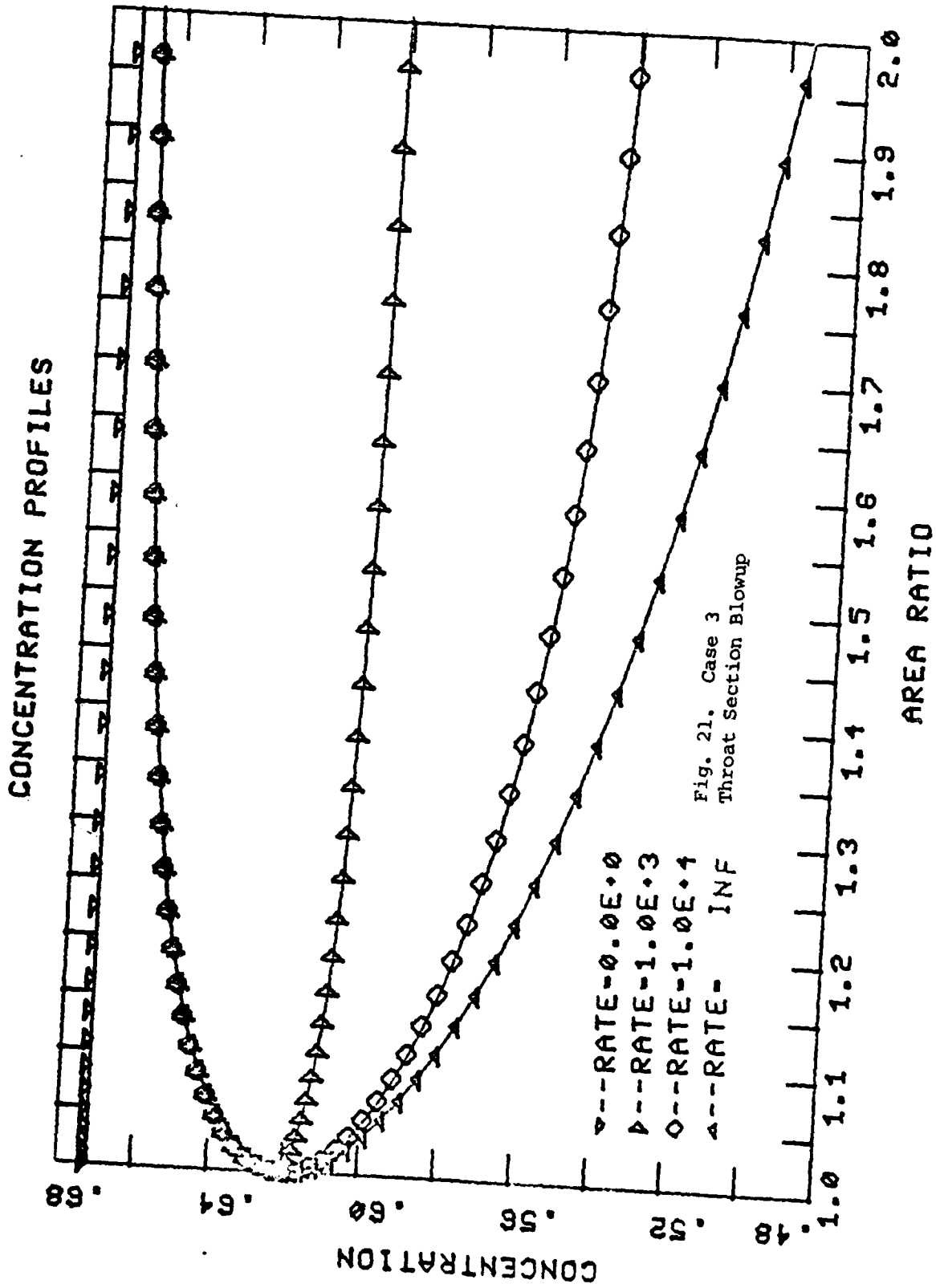


DENSITY PROFILES

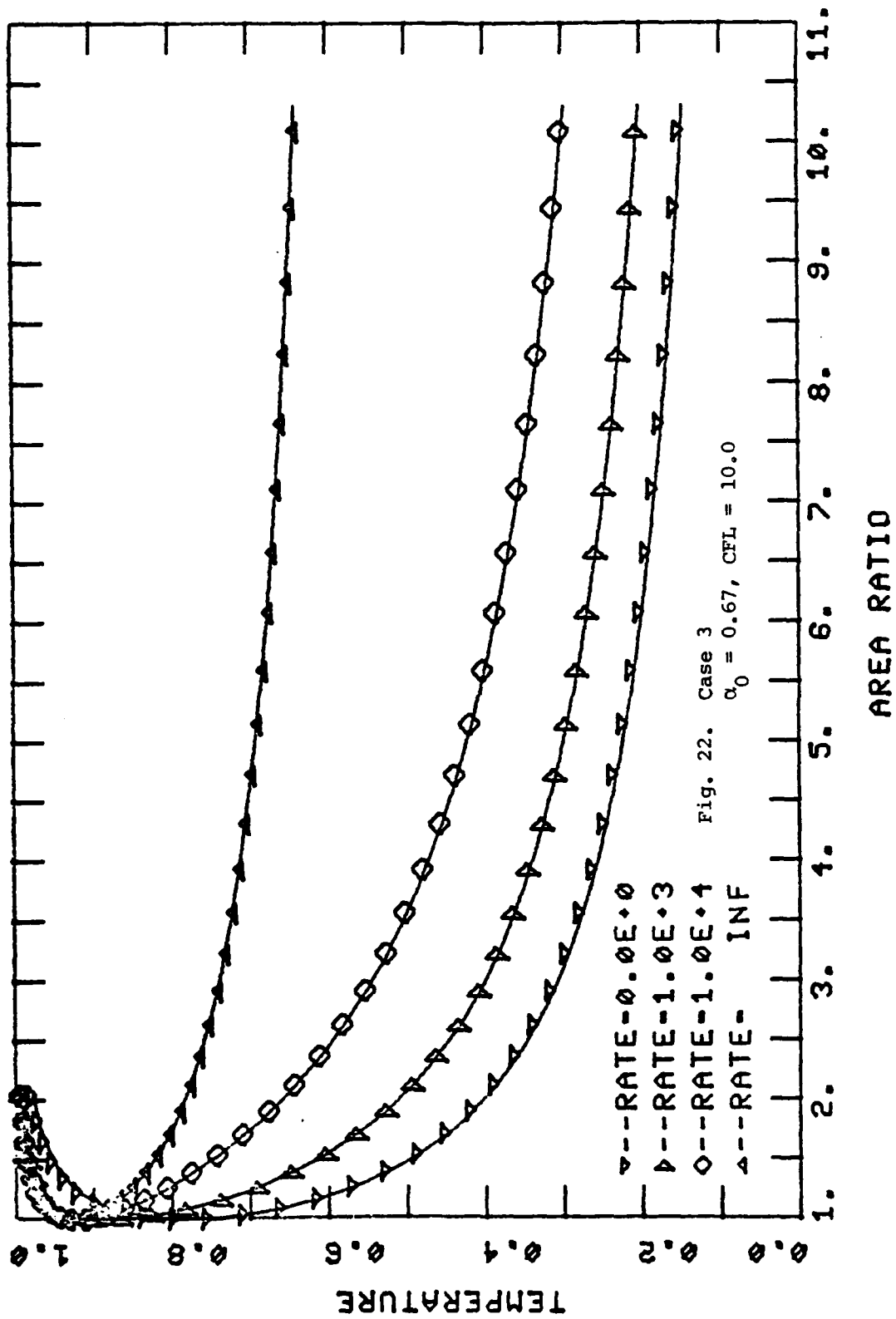


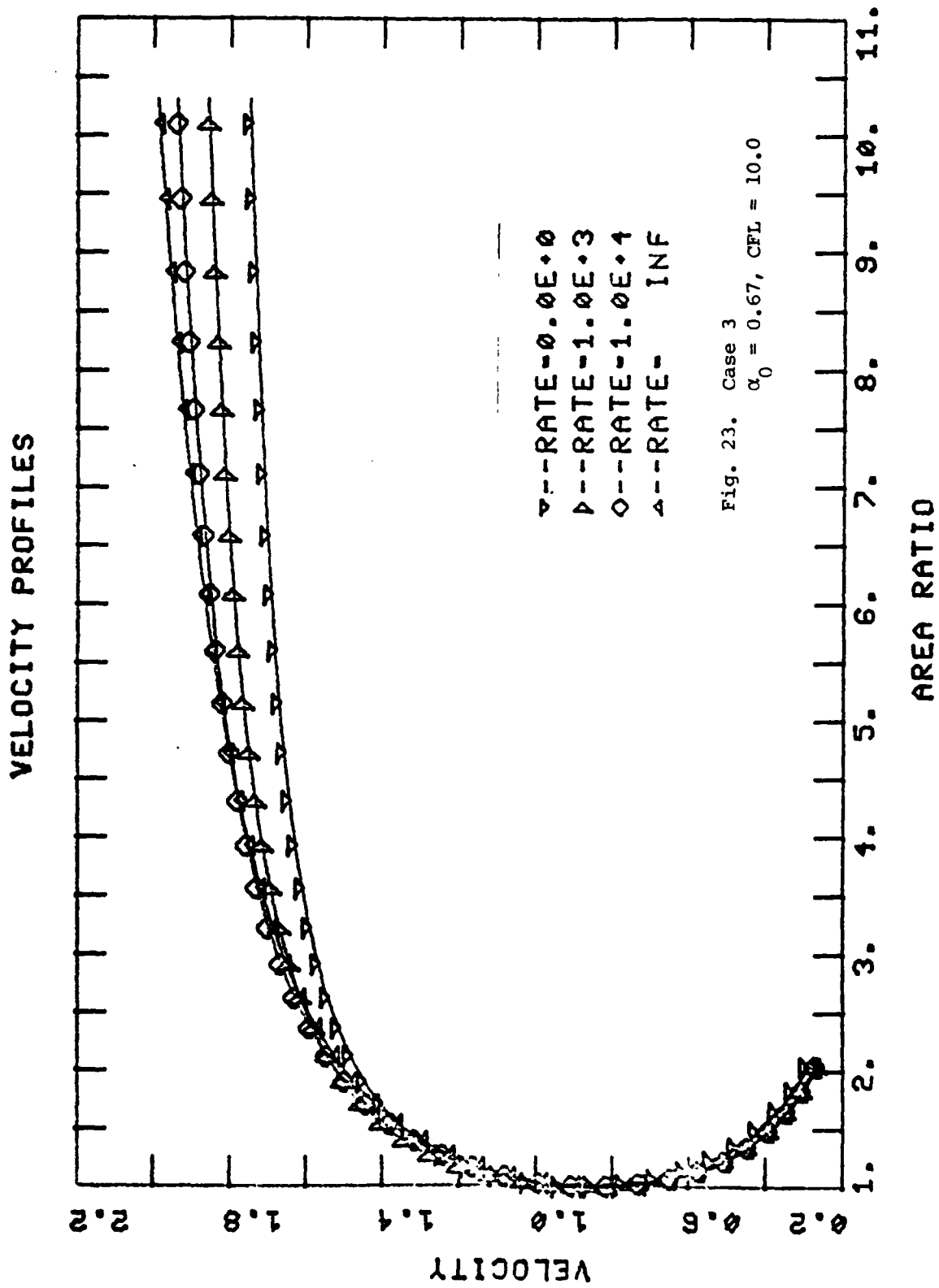
CONCENTRATION PROFILES



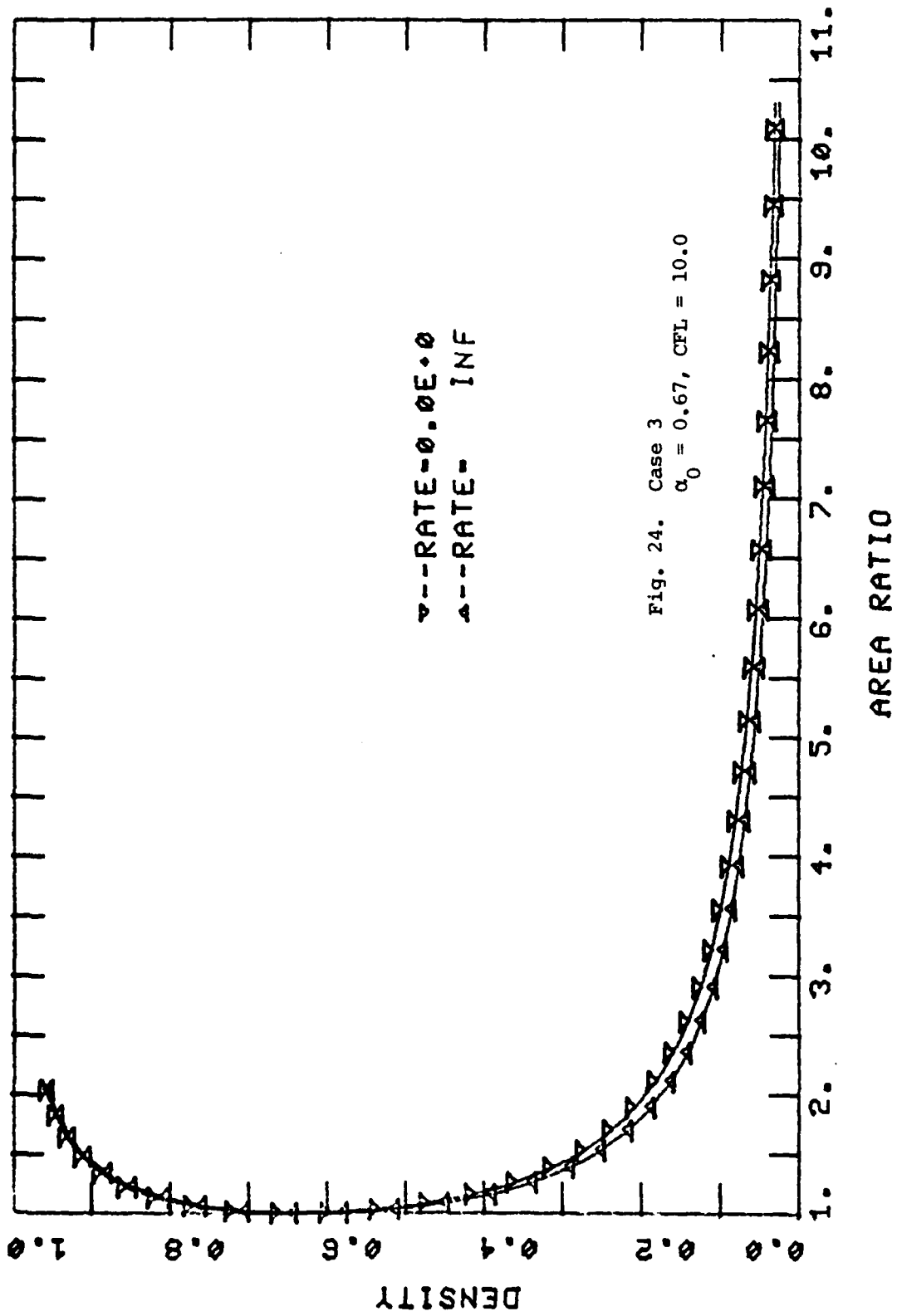


TEMPERATURE PROFILES

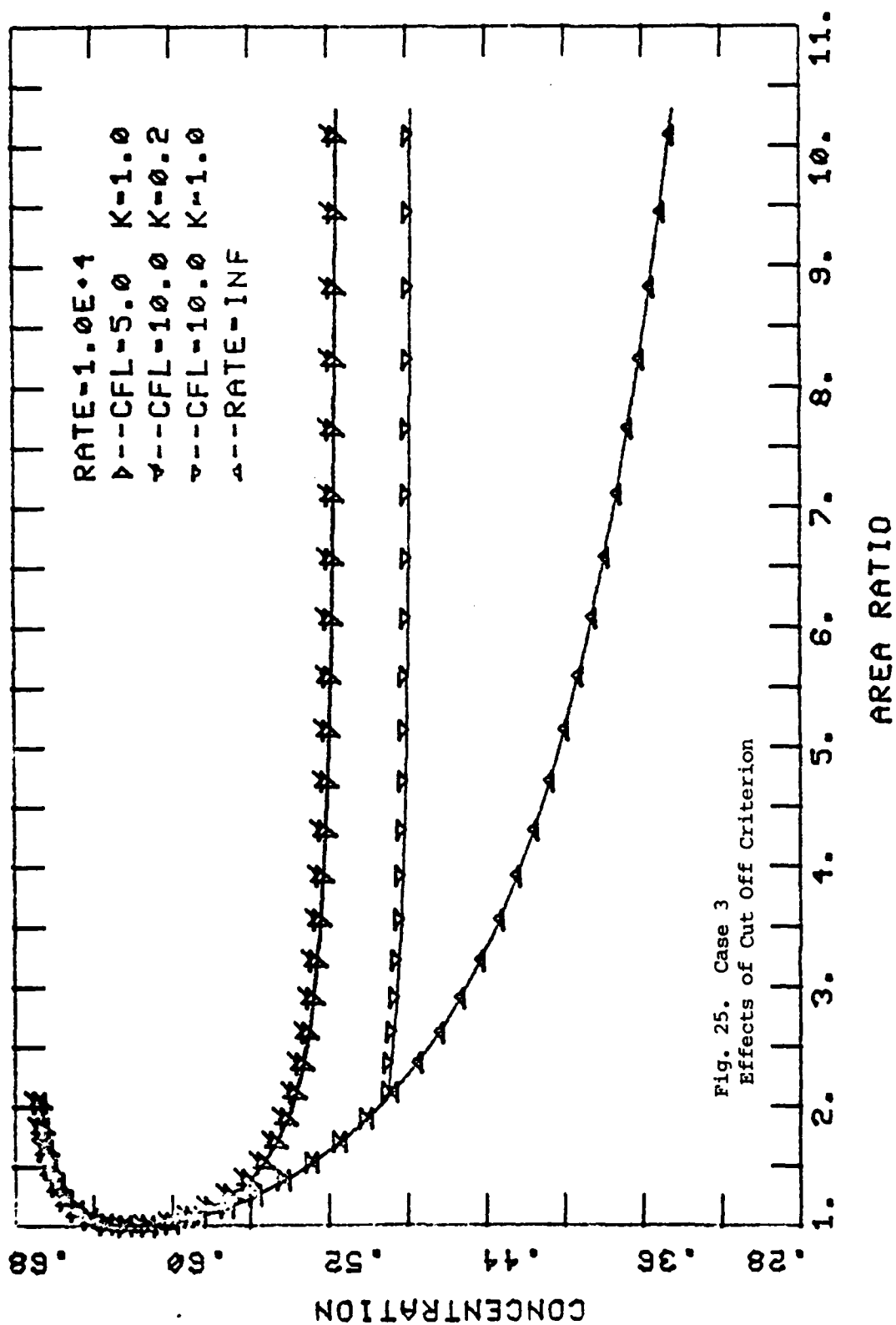




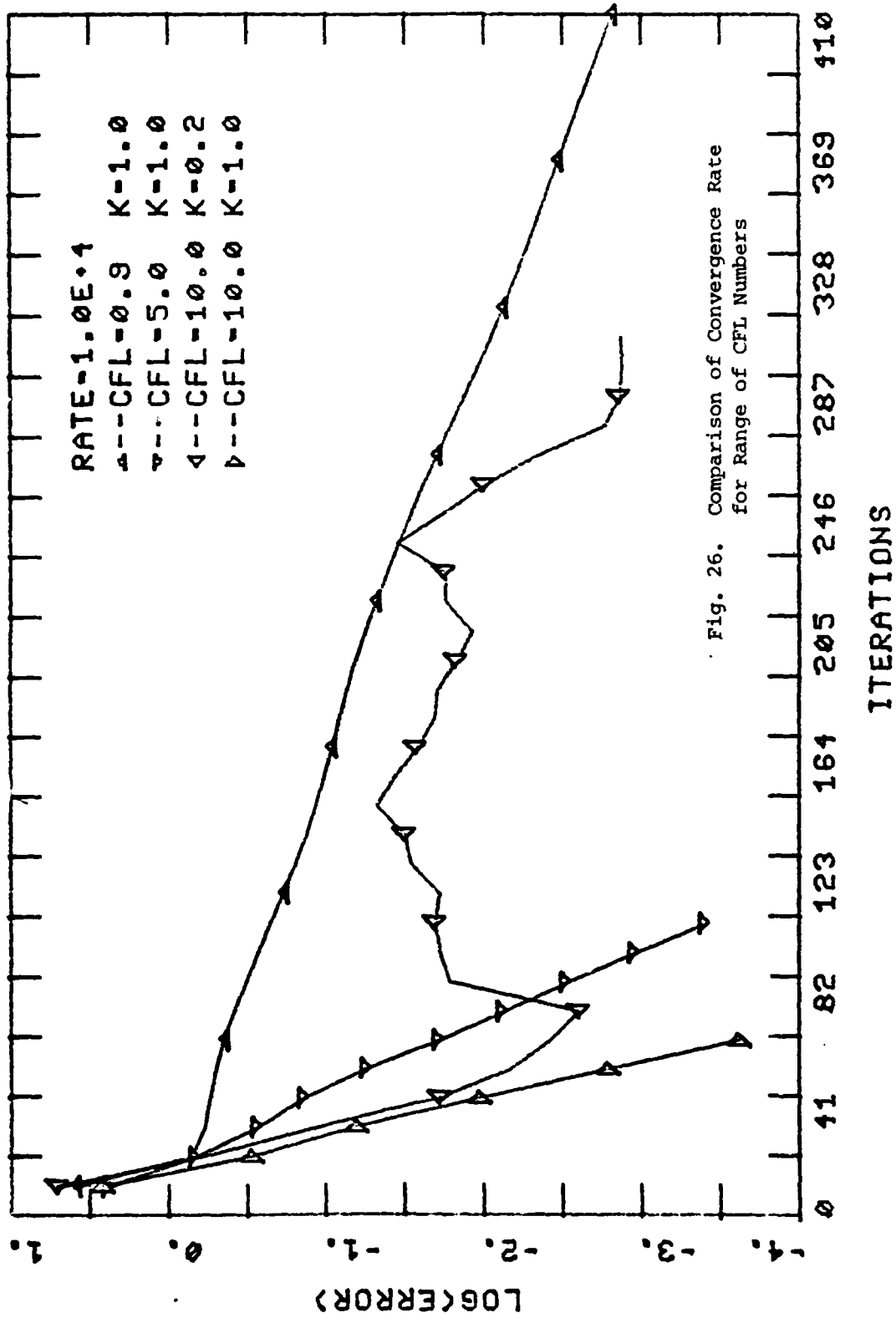
DENSITY PROFILES



EFFECTS OF CUT OFF CONDITION



ERROR HISTORY



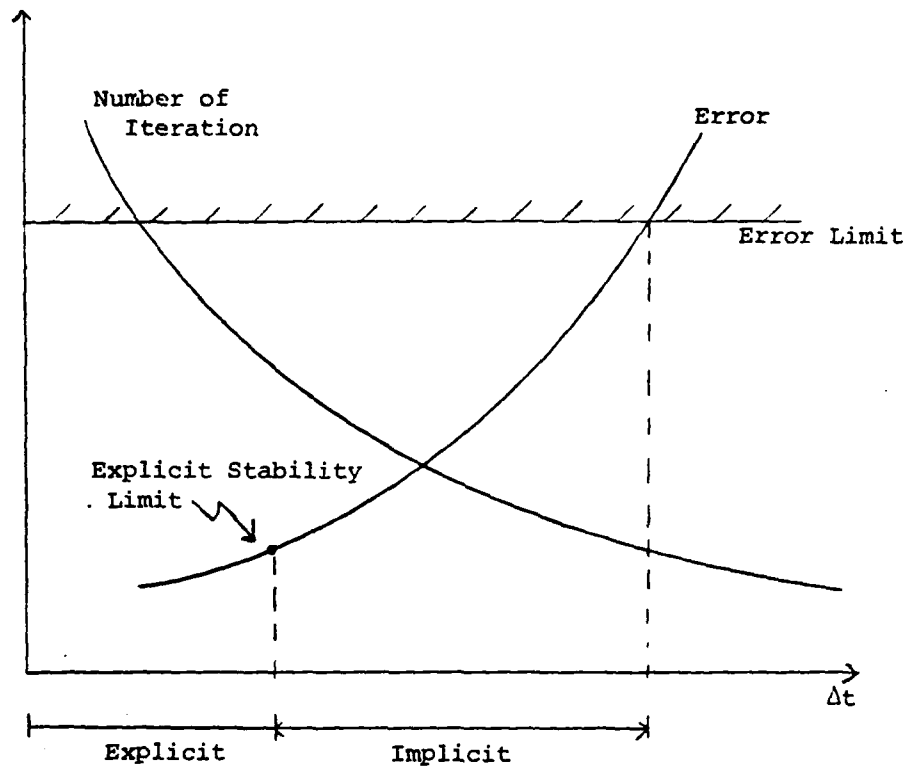
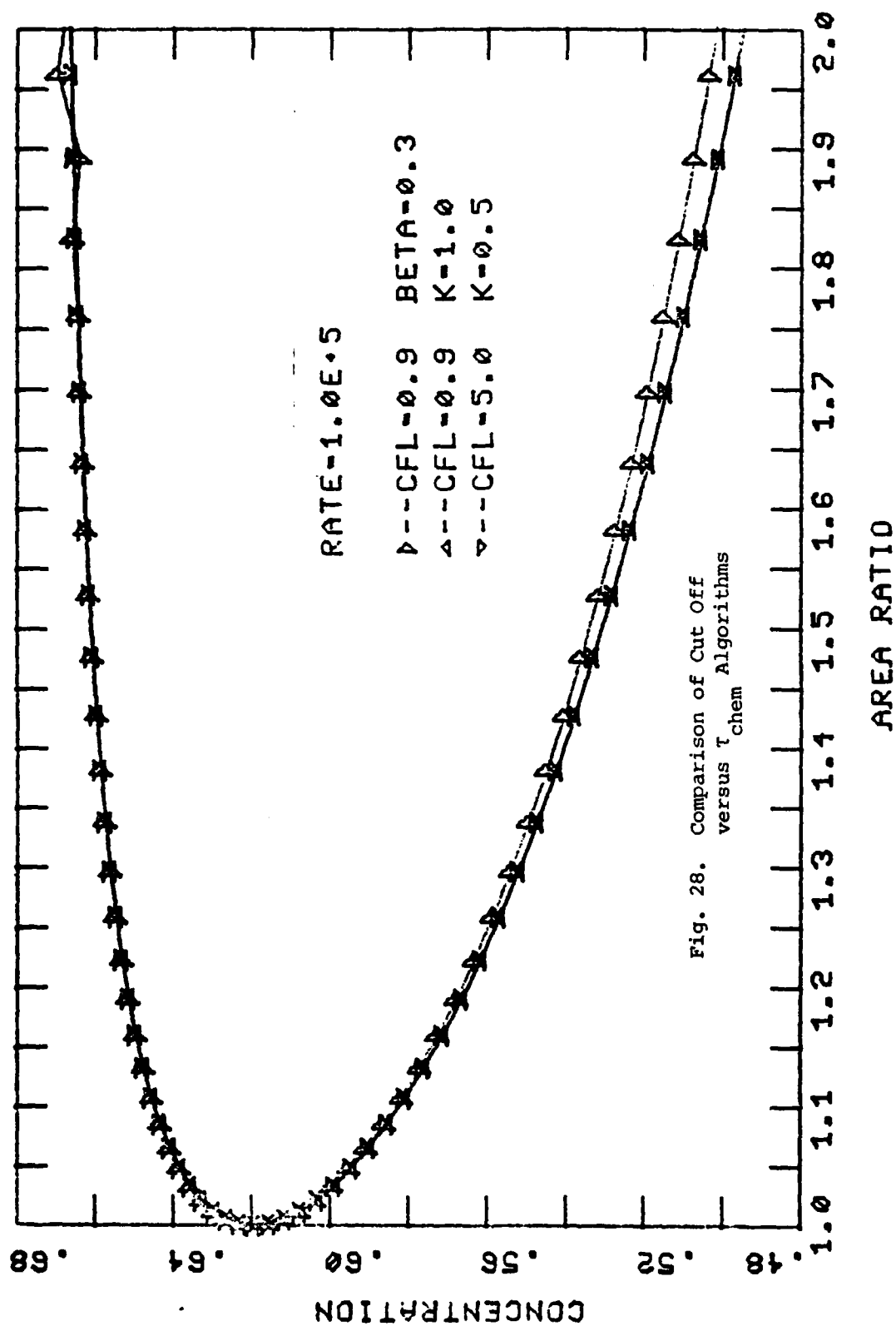


Fig. 27. Trends in Algorithm Accuracy and Iteration Count

CONCENTRATION PROFILES



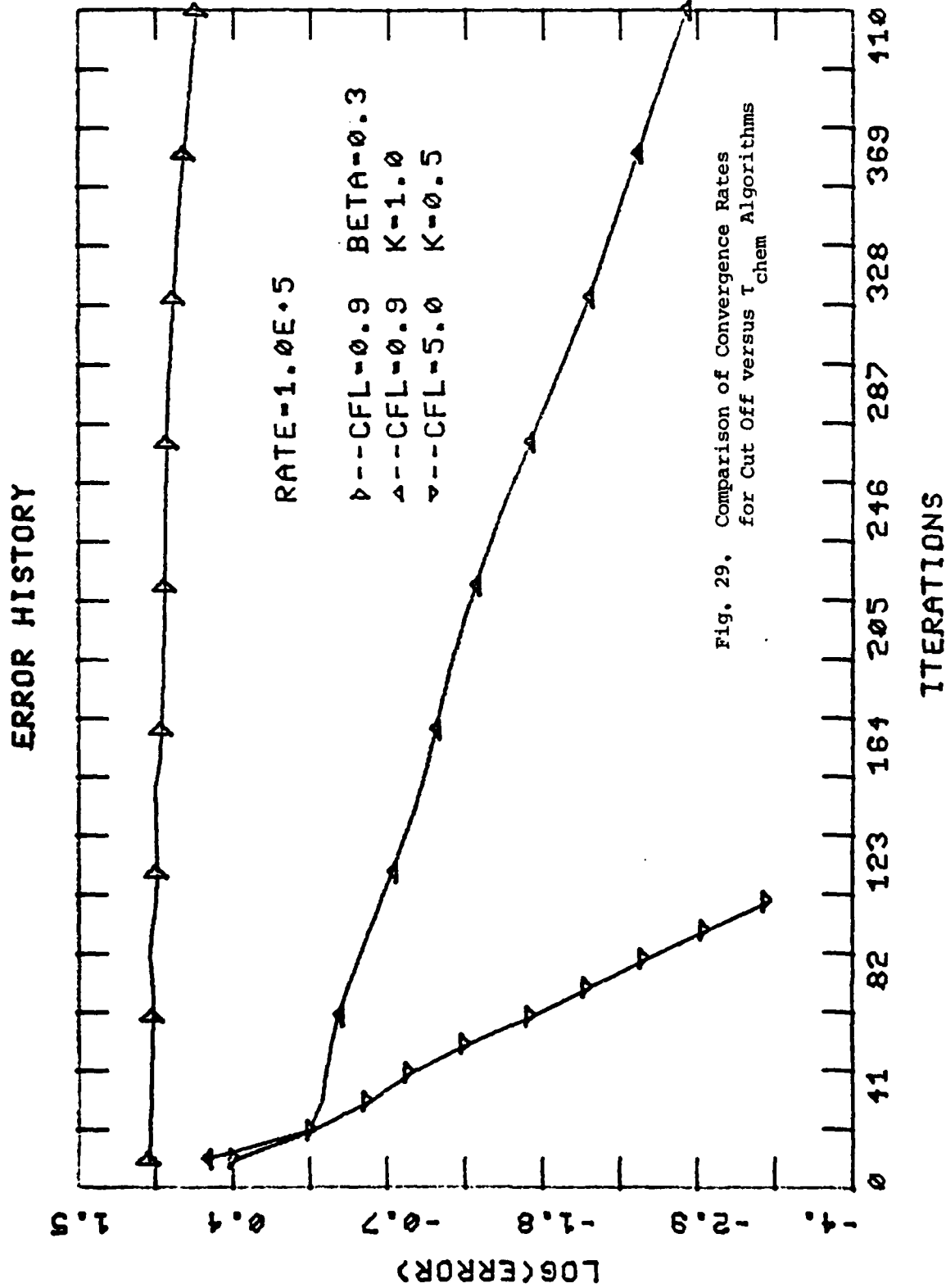
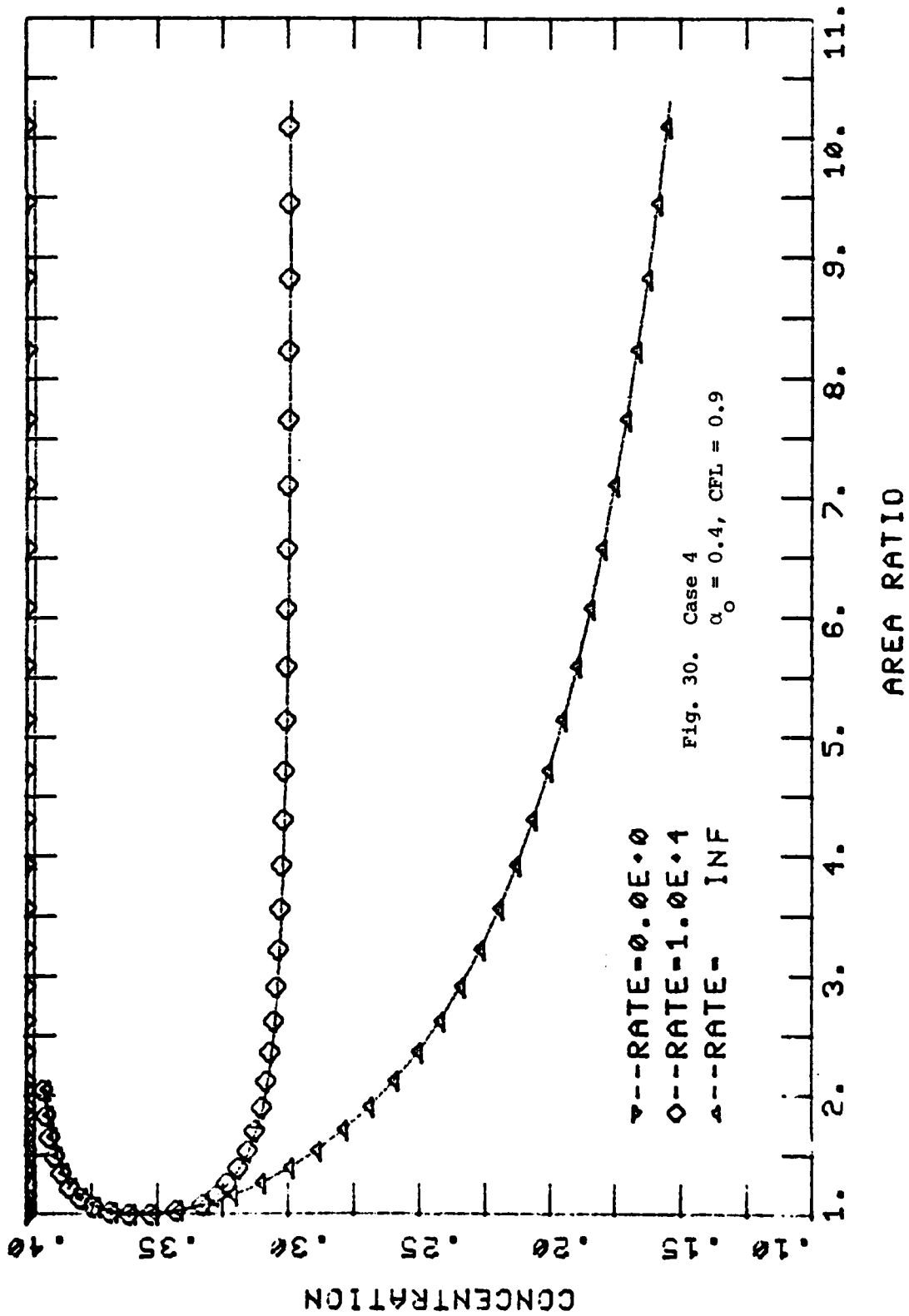
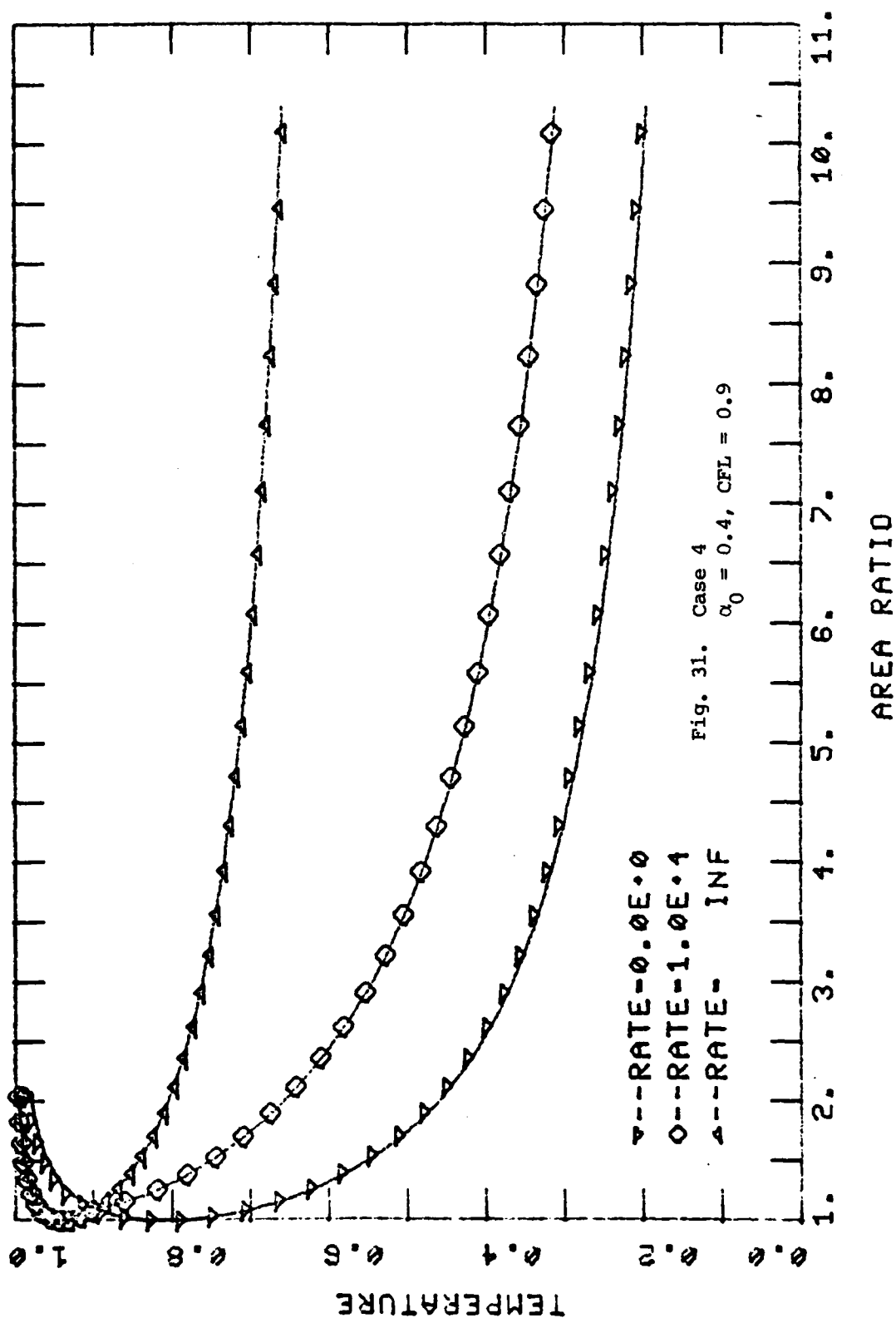


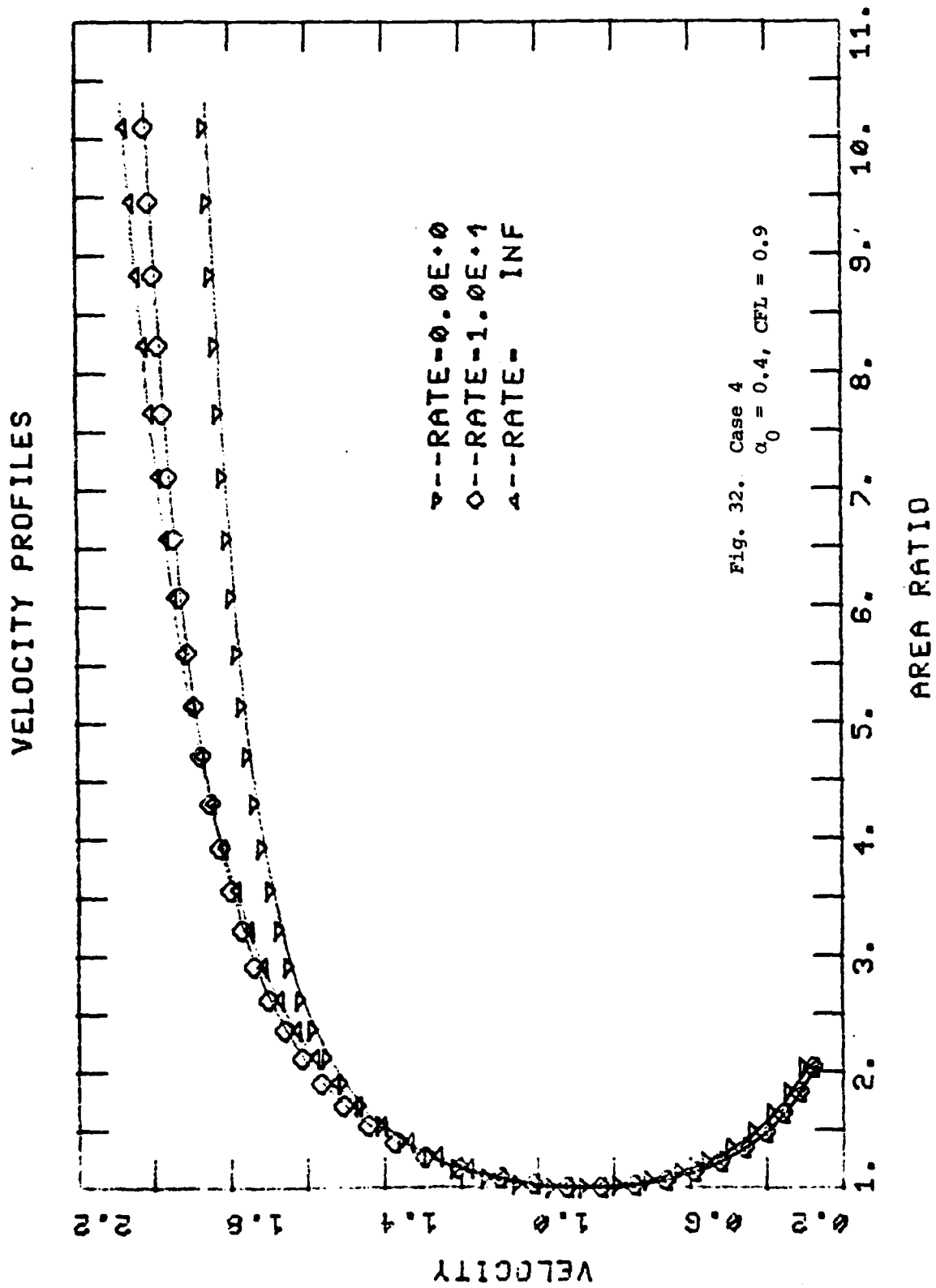
Fig. 29. Comparison of Convergence Rates
for Cut Off versus t_{chem} Algorithms

CONCENTRATION PROFILES

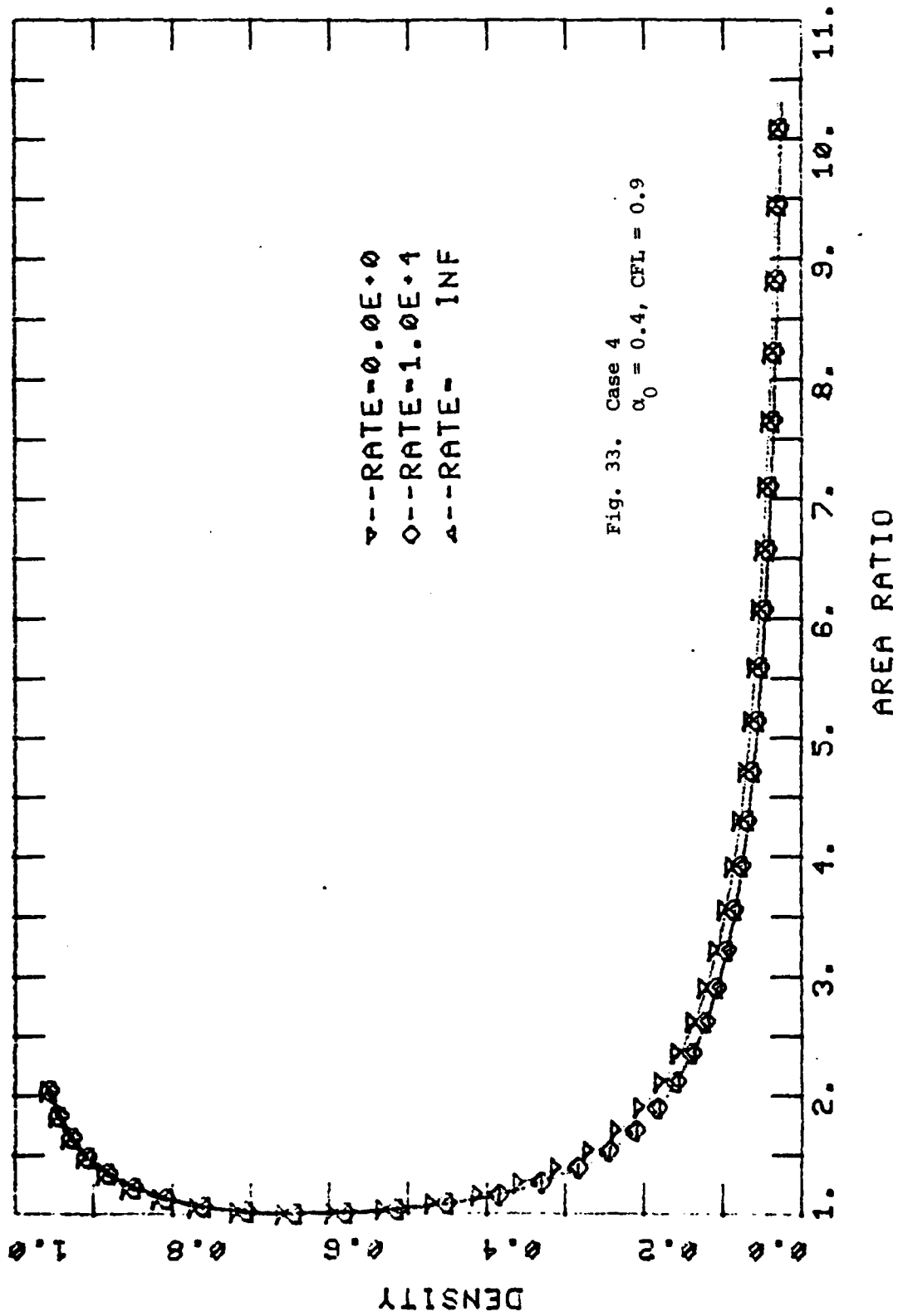


TEMPERATURE PROFILES

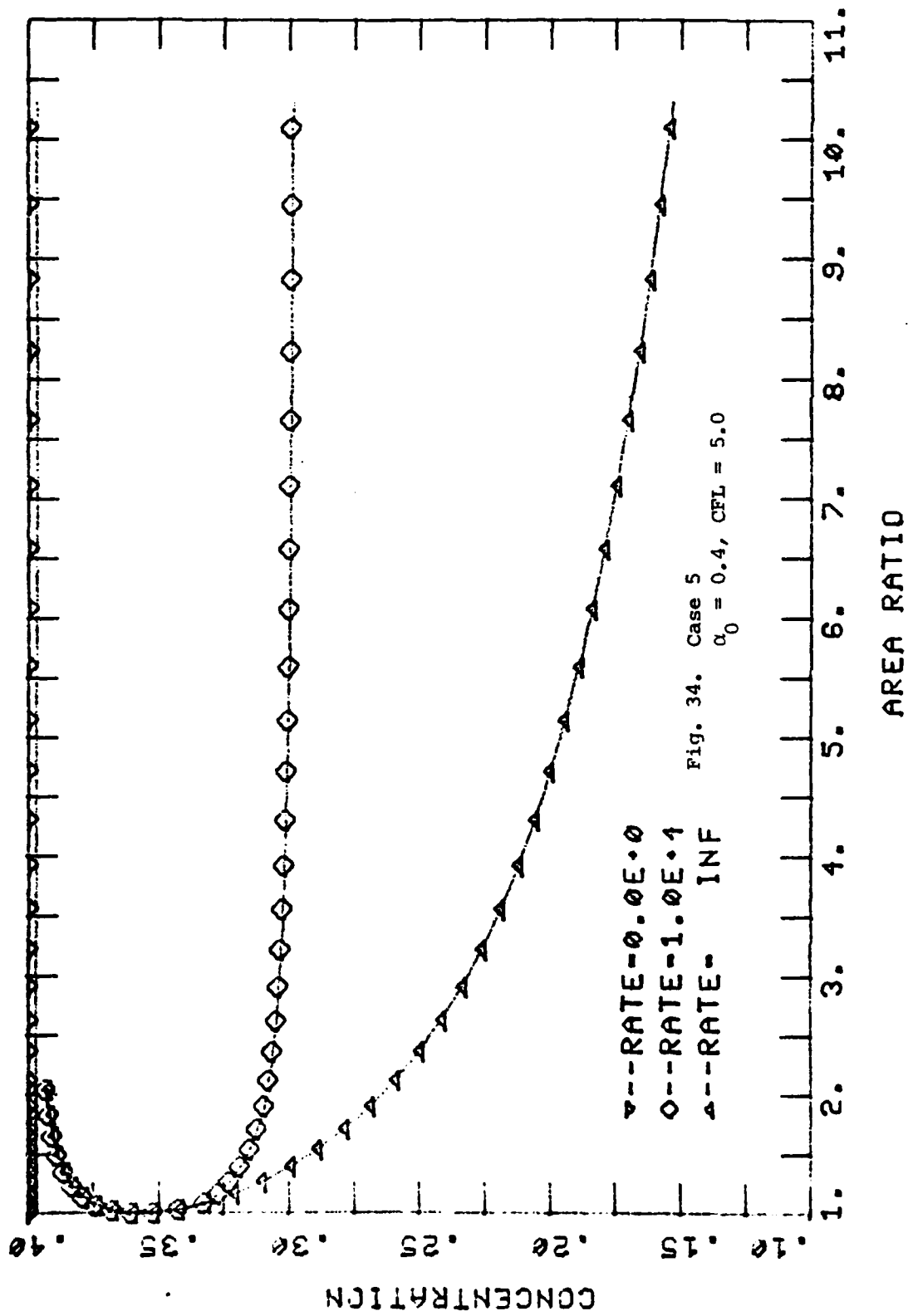




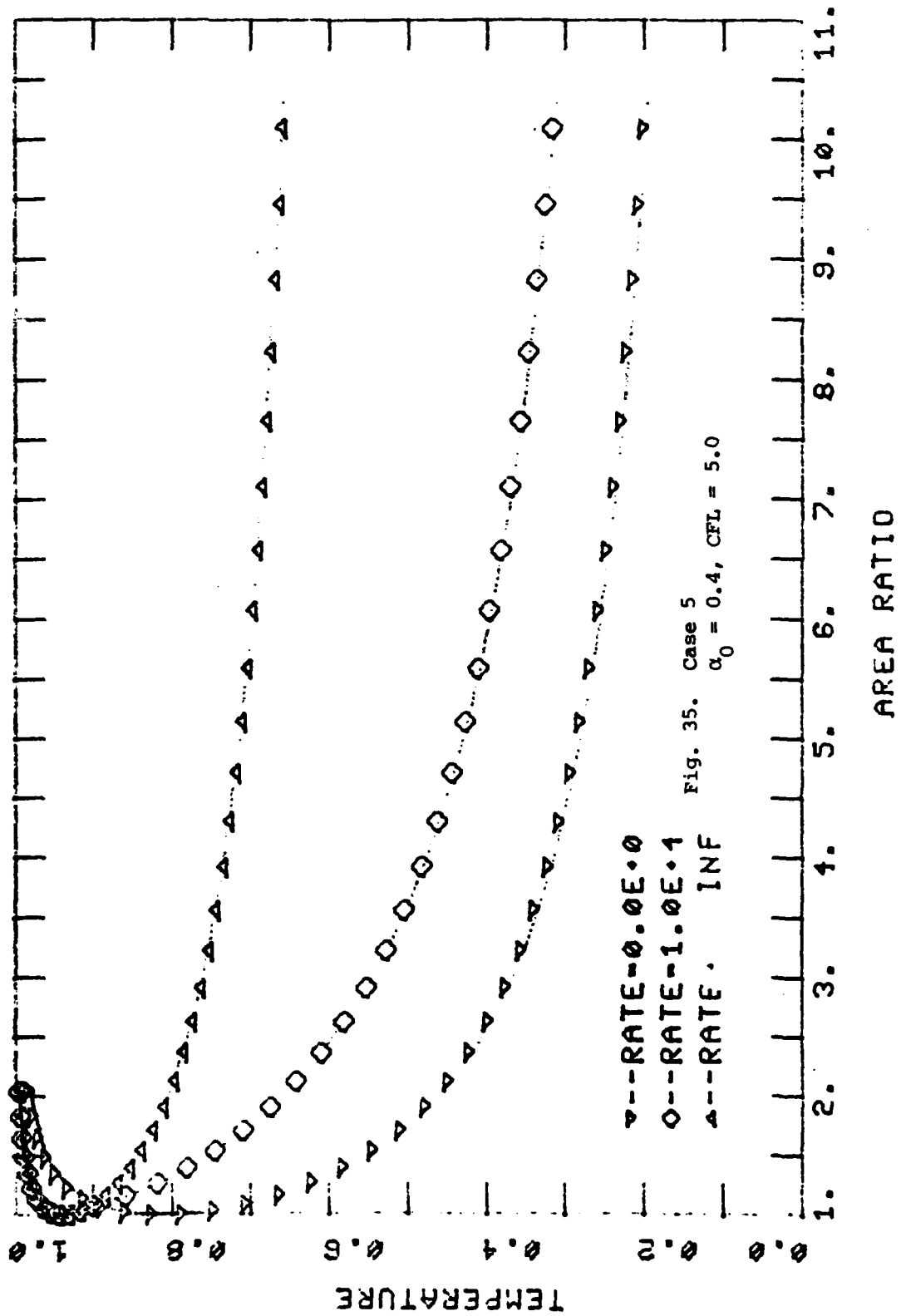
DENSITY PROFILES



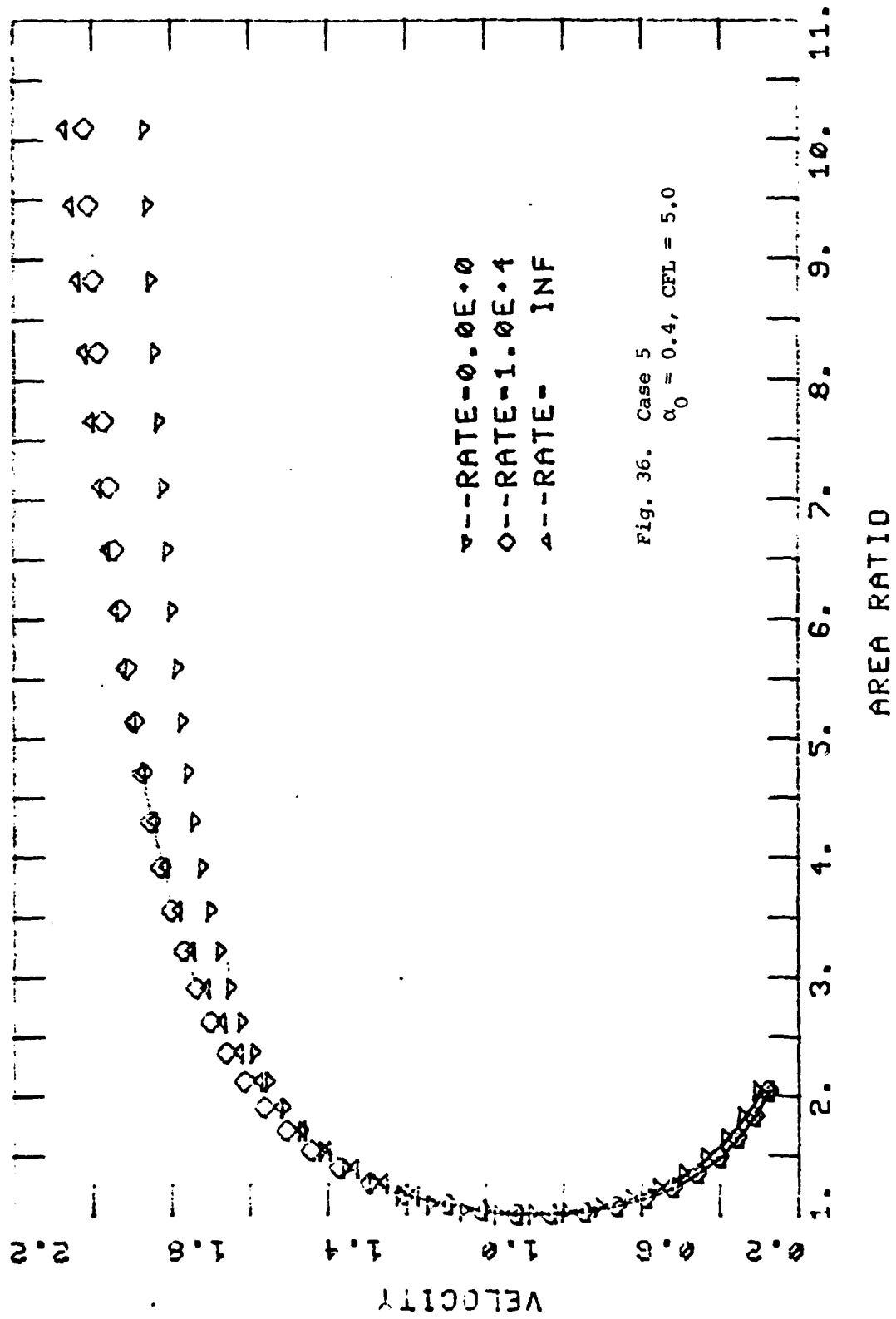
CONCENTRATION PROFILES



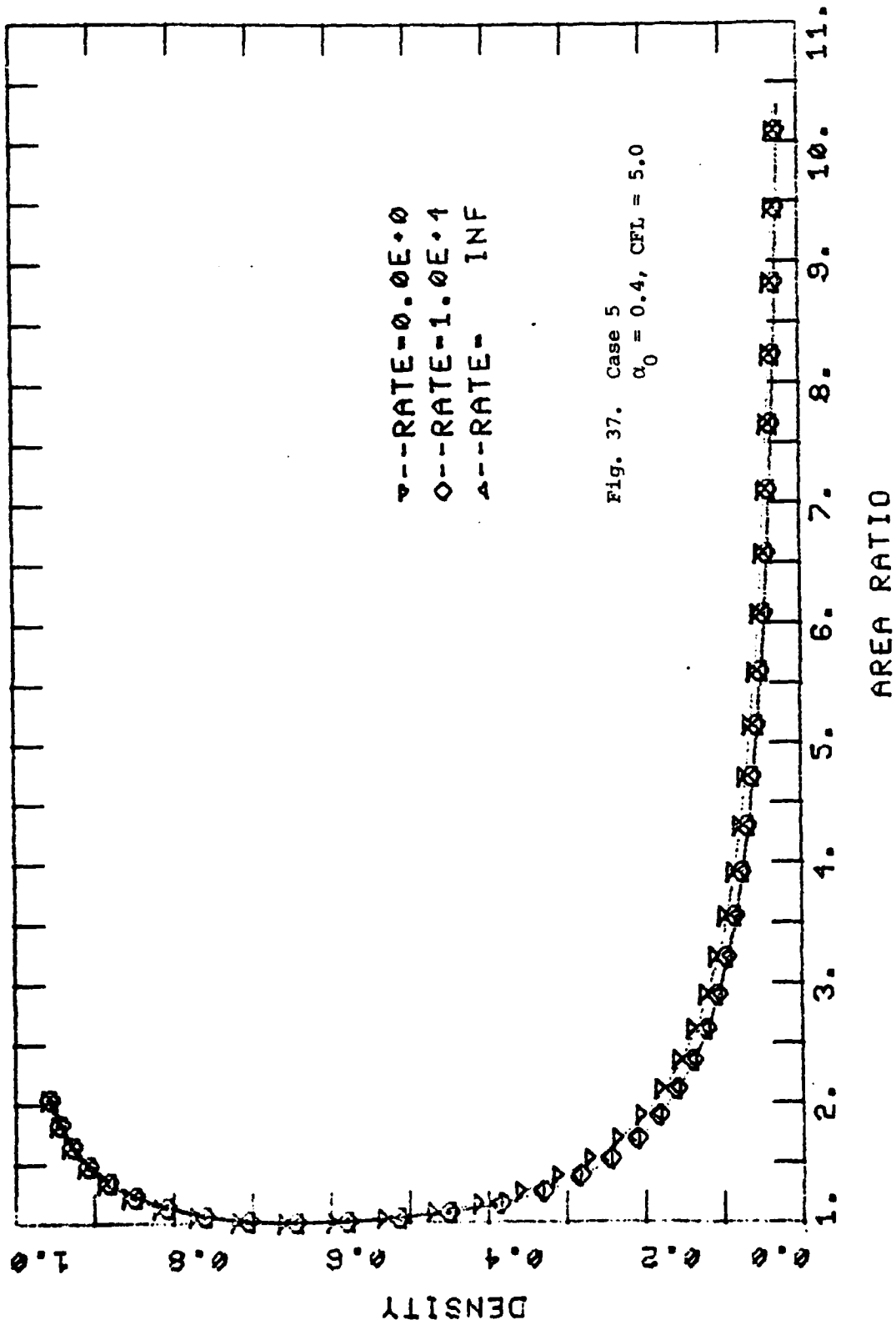
TEMPERATURE PROFILES



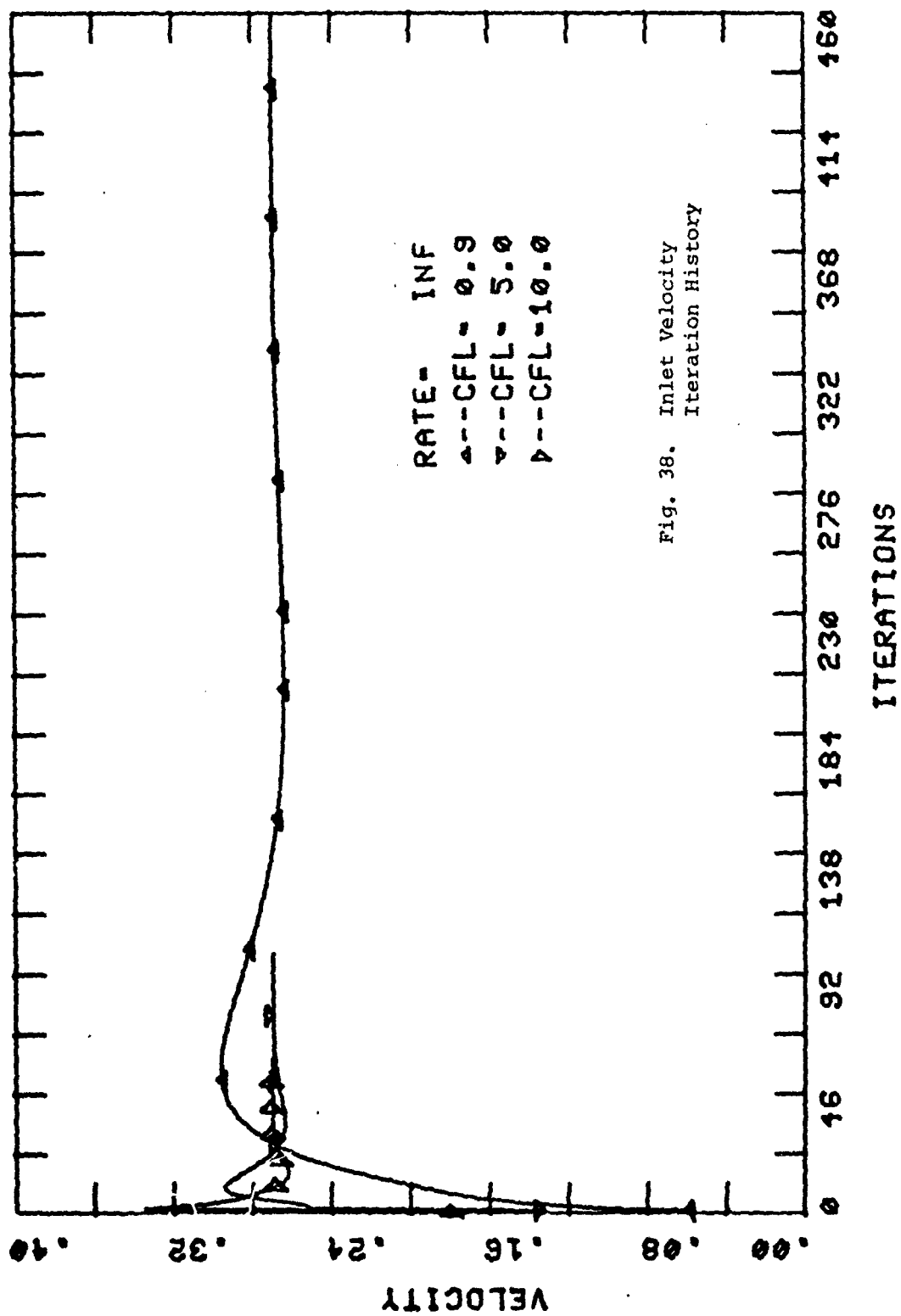
VELOCITY PROFILES

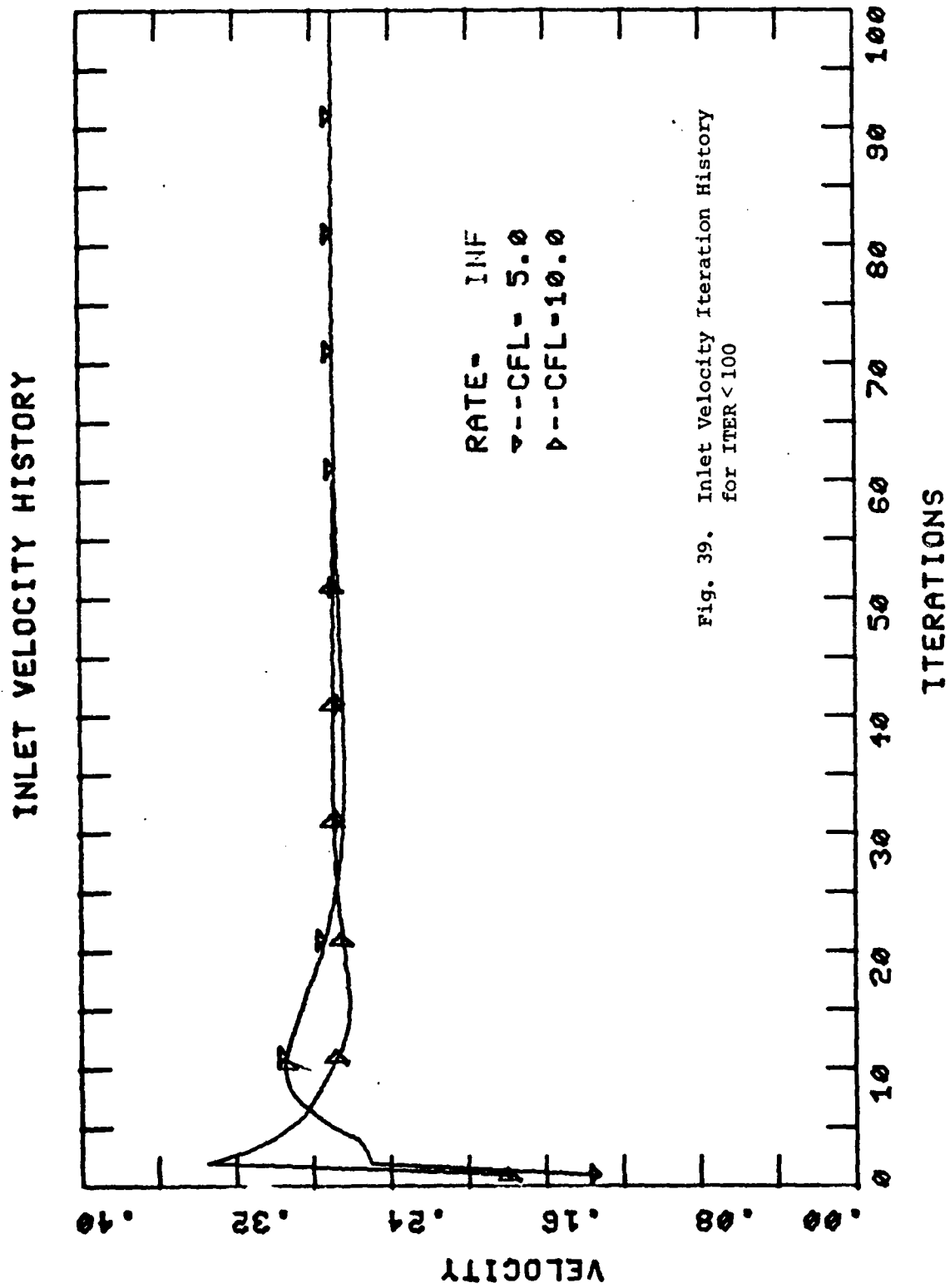


DENSITY PROFILES



INLET VELOCITY HISTORY

Fig. 38. Inlet Velocity
Iteration History



AD-A126 907

NUMERICAL INTEGRATION OF NONEQUILIBRIUM INTERNAL FLOW
USING UNSTEADY EULER EQUATIONS(U) AIR FORCE INST OF
TECH WRIGHT-PATTERSON AFB OH M J TROUT FEB 83

20

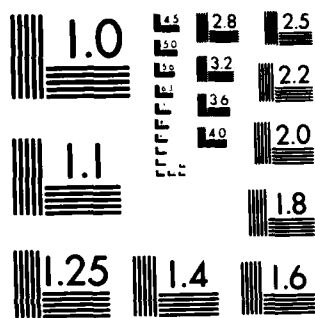
UNCLASSIFIED

AFIT/CI/NR-83-87

F/G 20/4

NL

END
DATE
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Appendix A

EIGENVALUES, EIGENVECTORS AND BOUNDARY CONDITIONS

Diagonalization of Euler Equation

As noted in Chapter 3 the system of nonequilibrium unsteady Euler equation can be solved using a block bidiagonal implicit scheme. In order to implement this algorithm, eigenvalues and eigenvectors of the system must be computed. This section will cover their derivation following procedures similar to Reference [24].

Nonequilibrium Case

For the nonequilibrium case the system of equations is:

$$(A1) \quad \begin{bmatrix} \rho \\ u \\ T \\ \alpha \end{bmatrix}_t + \begin{bmatrix} u & p & 0 & 0 \\ \frac{RT}{\epsilon_0 R_0} & u & \frac{R}{\epsilon_0 R_0} & \frac{R\alpha T}{\epsilon_0 R_0} \\ 0 & \frac{RT}{c_v} & u & 0 \\ 0 & 0 & 0 & u \end{bmatrix} \begin{bmatrix} \rho \\ u \\ T \\ \alpha \end{bmatrix}_x + \begin{bmatrix} \frac{\rho u}{A} \frac{\partial A}{\partial x} \\ 0 \\ \frac{RT}{c_v} \frac{u}{A} \frac{\partial A}{\partial x} + q \\ -\dot{w} \end{bmatrix}$$

or

$$(A2) \quad U_t + A U_x + H = 0$$

To diagonalize this set of equations the eigenvalues and eigenvector of A must be found from the relations

$$(A3) \quad \vec{X}_i A = \lambda_i \vec{X}_i$$

or

$$(A4) \quad \vec{x}_i (A - \lambda_i I) = \vec{0}$$

where x_i and λ_i are the row eigenvector and eigenvalues of matrix A.

The eigenvalues are computed by:

$$(A5) \quad \det (A - \lambda_i I) = 0$$

which becomes

$$(A6) \quad (\lambda - u)^2 (\lambda - (u + a)) (\lambda - (u - a)) = 0$$

Here $a^2 = \frac{(R + C_v)}{\gamma_0 R_0 C_v} RT$ which is the ratio of the local frozen speed of

sound to the stagnation speed of sound. Hence the eigenvalues are

$$(A7) \quad \begin{aligned} \lambda_1 &= u - a \\ \lambda_2 &= u \\ \lambda_3 &= u + a \\ \lambda_4 &= u \end{aligned}$$

The eigenvalues are calculated by substituting the eigenvectors into

Equation (A.5)

$$(A8) \quad \begin{aligned} \lambda_1 &= \left(1, -\frac{a \gamma_0 R_0 \rho}{RT}, \frac{\rho}{T}, \frac{R_{A_2} \rho}{R} \right) \\ \lambda_2 &= \left(1, 0, -\frac{\rho C_v}{RT}, 0 \right) \\ \lambda_3 &= \left(1, \frac{a \gamma_0 R_0 \rho}{RT}, \frac{\rho}{T}, \frac{R_{A_2} \rho}{R} \right) \\ \lambda_4 &= \left(0, 0, 0, 1 \right) \end{aligned}$$

These eigenvectors form the rows of a matrix S_x . Now the matrix A can be diagonalized by the similarity transform

$$(A9) \quad \Delta = S_x A S_x^{-1}$$

where Λ is the diagonal matrix of eigenvectors

$$(A10) \quad S_x = \begin{bmatrix} 1 & -\frac{a \gamma_0 R_0 p}{RT} & \frac{p}{T} & \frac{R_{A_2} p}{R} \\ 1 & 0 & -\frac{p C_v}{RT} & 0 \\ 1 & \frac{a \gamma_0 R_0 p}{RT} & \frac{p}{T} & \frac{R_{A_2} p}{R} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(A11) \quad S_x^{-1} = \frac{1}{R + C_v} \begin{bmatrix} \frac{C_v}{2} & R & \frac{C_v}{2} & -\frac{p C_v R_{A_2}}{R} \\ -\frac{RT(R+C_v)}{2a\gamma_0 R_0 p} & 0 & \frac{RT(R+C_v)}{2a\gamma_0 R_0 p} & 0 \\ \frac{RT}{2p} & -\frac{RT}{p} & \frac{RT}{2p} & -R_{A_2} T \\ 0 & 0 & 0 & (R+C_v) \end{bmatrix}$$

$$(A12) \quad \Lambda = \begin{bmatrix} u-a & & & \\ & u & & \\ & & u+a & \\ & & & u \end{bmatrix}$$

Equilibrium Case

For the equilibrium system of equations:

$$(A13) \quad \begin{bmatrix} \rho \\ u \\ T \end{bmatrix}_t + \begin{bmatrix} u & p & 0 \\ \left(\frac{R_{A2} T}{\gamma_0 R_0} \frac{\partial \alpha}{\partial p} + \frac{RT}{\gamma_0 R_0 p} \right) u & \left(\frac{R}{\gamma_0 R_0} + \frac{R_{A2} T}{\gamma_0 R_0} \frac{\partial \alpha}{\partial T} \right) & \\ 0 & k_1 \left(\frac{RT}{C_p} - k_2 p \right) & u \end{bmatrix} \cdot \begin{bmatrix} \rho \\ u \\ T \end{bmatrix}_x + \begin{bmatrix} \frac{\rho u}{A} \frac{\partial A}{\partial x} \\ 0 \\ k_1 \left(\frac{RT}{C_p} - k_2 p \right) \frac{u}{A} \frac{\partial A}{\partial x} \end{bmatrix} = 0$$

where $k_1 = \frac{1}{1 + \frac{\Theta_d}{\gamma_0} \frac{\partial \alpha}{\partial T}}$, $k_2 = \frac{\Theta_d}{3} \frac{\partial \alpha}{\partial p}$

By the same procedure, the eigenvalues of A are here

$$(A14) \quad \begin{aligned} \lambda_1 &= u - a \\ \lambda_2 &= u \\ \lambda_3 &= u + a \end{aligned}$$

where

$$a^2 = \frac{R_{A2} T [\alpha (1-\alpha)^2 (1 + 2T/\Theta_d) + (8 + 3\alpha - \alpha^3) (T/\Theta_d)^2]}{\gamma_0 R_0 [\alpha (1-\alpha) + 3(2-\alpha) (T/\Theta_d)^2]}$$

which is the ratio of the local equilibrium sound speed to the stagnation sound speed.

The matrix of eigenvalues is then computed using (A.5)

$$(A15) \quad S_x = \begin{bmatrix} 1 & \frac{-\alpha \gamma_0 R_0 p}{R_{A2} T p \frac{\partial \alpha}{\partial p} + RT} & \frac{p}{T} \left[\frac{R + R_{A2} T \frac{\partial \alpha}{\partial T}}{R + R_{A2} p \frac{\partial \alpha}{\partial p}} \right] \\ 1 & 0 & \frac{-p}{K_1 \left(\frac{RT}{C_v} - K_2 p \right)} \\ 1 & \frac{\alpha \gamma_0 R_0 p}{R_{A2} T p \frac{\partial \alpha}{\partial p} + RT} & \frac{p}{T} \left[\frac{R + R_{A2} T \frac{\partial \alpha}{\partial T}}{R + R_{A2} p \frac{\partial \alpha}{\partial p}} \right] \end{bmatrix}$$

Boundary Conditions

The implicit boundary condition at the subsonic inlet is applied as in Reference [22]. For the case of equilibrium flow there are two positive and one negative eigenvalue. Therefore two boundary conditions can be applied:

$$S = \text{constant}$$

$$h_0 = \text{constant}$$

For an ideal dissociating gas

$$(A16) \quad S = 3 \ln \frac{T}{\Theta_d} + \alpha(1 - 2 \ln \alpha) - (1 - \alpha) \ln(1 - \alpha) \\ - (1 + \alpha) \ln \frac{p}{p_d} + \text{const}$$

$$(A17) \quad h_0 = R_{A2} \left[(4 + \alpha) T + \alpha \Theta_d \right] + \frac{u^2}{2}$$

Applying these boundary conditions in time linear form

$$(A18) \quad \frac{\partial S}{\partial u} \frac{\partial u}{\partial t} = 0, \quad \frac{\partial h_0}{\partial u} \frac{\partial u}{\partial t} = 0$$

Therefore,

$$(A19) \quad \frac{\partial S}{\partial p} = -\frac{(1+\alpha)}{p} + \frac{\Theta_d}{T} \frac{\partial \alpha}{\partial p}$$

$$(A20) \quad \frac{\partial S}{\partial T} = \frac{3}{T} + \frac{\Theta_d}{T} \frac{\partial \alpha}{\partial T}$$

$$(A21) \quad \frac{\partial h_0}{\partial p} = R_{A_2} (T + \Theta_d) \frac{\partial \alpha}{\partial p}$$

$$(A22) \quad \frac{\partial h_0}{\partial u} = u$$

$$(A23) \quad \frac{\partial h_0}{\partial T} = R_{A_2} (4 + \alpha) + R_{A_2} (T + \Theta_d) \frac{\partial \alpha}{\partial T}$$

By eliminating the eigenvectors of the positive eigenvalues and replacing them with the boundary conditions

$$(A24) \quad P_1 = \begin{bmatrix} 1 & -\frac{\alpha \gamma_0 R_0 p}{RT} & \frac{p}{T} \\ (T + \Theta_d) \frac{\partial \alpha}{\partial p} & \frac{(4 + \alpha)(1 + \alpha)u}{3} & (4 + \alpha) + (T + \Theta_d) \frac{\partial \alpha}{\partial T} \\ -\frac{(1 + \alpha)}{p} + \frac{\Theta_d}{T} \frac{\partial \alpha}{\partial p} & 0 & \frac{3}{T} + \frac{\Theta_d}{T} \frac{\partial \alpha}{\partial T} \end{bmatrix}$$

$$(A25) \quad P_2 = \begin{bmatrix} 1 & -\frac{\alpha \gamma_0 R_0 p}{RT} & \frac{p}{T} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Therefore, the system of equations at the boundary becomes

$$(A26) \quad P_1 U_t + P_2 A U_x + P_2 H = 0$$

or

$$(A27) \quad U_t = -P_1^{-1} P_2 (A U_x + H)$$

The explicit/implicit algorithm is used at this point with forward differences only. At the supersonic outlet all eigenvalues are positive thus no special treatment is necessary.

Algorithm Implementation

With the eigenvalues, eigenvectors and boundary conditions specific, the implementation of the implicit scheme is as follows:

Recall that the bidiagonal matrix equation is

$$(A28) \quad \left(I + \frac{\Delta t}{\Delta x} |A|_i^n \right) \delta U_i^{n+1} = \Delta U_i^n + \frac{\Delta t}{\Delta x} |A|_i^n \delta U_{i+1}^{n+1}$$

Rather than invert the matrix

$$\left(I + \frac{\Delta t}{\Delta x} |A|_i^n \right)$$

The following matrix algebra may be performed

$$(A29) \quad S_x \left(I + \frac{\Delta t}{\Delta x} |A|_i \right) S_x^{-1} S_x \delta U_i^{n+1} = S_x W$$

or

$$\left(I + \frac{\Delta t}{\Delta x} D_A \right) S_x \delta U_i^{n+1} = S_x W$$

The diagonal matrix

$$\left(I + \frac{\Delta t D_A}{\Delta x} \right)$$

proves to be much more easily integrated.

The complete sequence is as follows: (Reference [17])

- 1) $W = \Delta U_i^n + \frac{\Delta t}{\Delta x} |A|_i^n \delta U_i^{n+1}$
- 2) $X = S_x W$
- 3) D_A calculated
- 4) $Y = \left(I + \frac{\Delta t}{\Delta x} D_A \right)^{-1} X$
- 5) $\delta U_i^{n+1} = S_x^{-1} Y$
- 6) $Z = D_A Y$
- 7) $|A| \delta U_i^{n+1} = S_x^{-1} Z$

Appendix B

COMPUTER PROGRAM INPUT VARIABLE NAMES

The following is the list of user supplied input variable names for the computer program. The definition of some geometric variables are illustrated in Figure B1. Dimensions are given where appropriate.

CAI	Species concentration at $i = 1$
CF	Chemical rate constant (Reference [13]) page 231)
CFL	Courant number
DELMAX	Convergence criteria (Equation (60))
ETA	Temperature exponent (Reference [13] page 231)
IBUG	Debug printout if = 1
ICHEM	Nonequilibrium if = 2 Equilibrium if = 3
ICON	Detail printout for specified mesh point if = 1
ICTOFF	Apply cut off criterion if = 1
IGEO	= 0 no minimum section = 1 minimum section
INLET	= 0 characteristic inlet boundary conditions = 1 fixed inlet boundary condition
IPT	Specify mesh point for detailed printout
IREAD	= 1, read initial conditions from unit 98
ITAPE	= 1, write dependent variables of final iteration to unit 99 = 2, write error history to unit 77
ITMAX	maximum number of iterations
IWRITE	interval for writing data to output file
NMAX	number of mesh points

P0	Stagnation pressure [Atm]
RD	Characteristic dissociation density ρ_d [g/cm ³]
RH01	Initial inlet density
RH02	Linear decay coefficient
THD	Characteristic dissociation temperature
TRLX	Reaction rate coefficient (ϕ)
T0	Stagnation temperature [$^{\circ}$ K]
T1	Initial inlet temperature
T2	Linear decay coefficient
V0	Initial inlet velocity
V1	Linear growth coefficient
XKCHEM	Cut off coefficient (K)
XL	Distance from minimum section to point where YB is specified (Figure A1)
XLT	Total length if IGEO = 1 length to exit from origin if IGEO = 0
XMA2	Molecular weight of diatomic species
YB	Half height at $X = XL$

All data entered is formatted. Real variables are read using F10.0 format, except for TRLX and CF which are read in E10.3 format. Integer variables (those beginning with I or N) are read with I5 format. The data entry sequence is as follows with each line representing a data card or line entry:

NMAX	ITMAX	IWRITE	IBUG	DELMAX	
XL	XLT	YB	CFL		
RH01	RH02	T1	T2	V0	V1
ICHEM	T0	P0	XMA2		
RD	THD	ETA	TRLX	CF	
ICON	IPT				
ICTOFF	IGEO	INLET	IREAD	ITAPE	
CAL	XKCHEM				

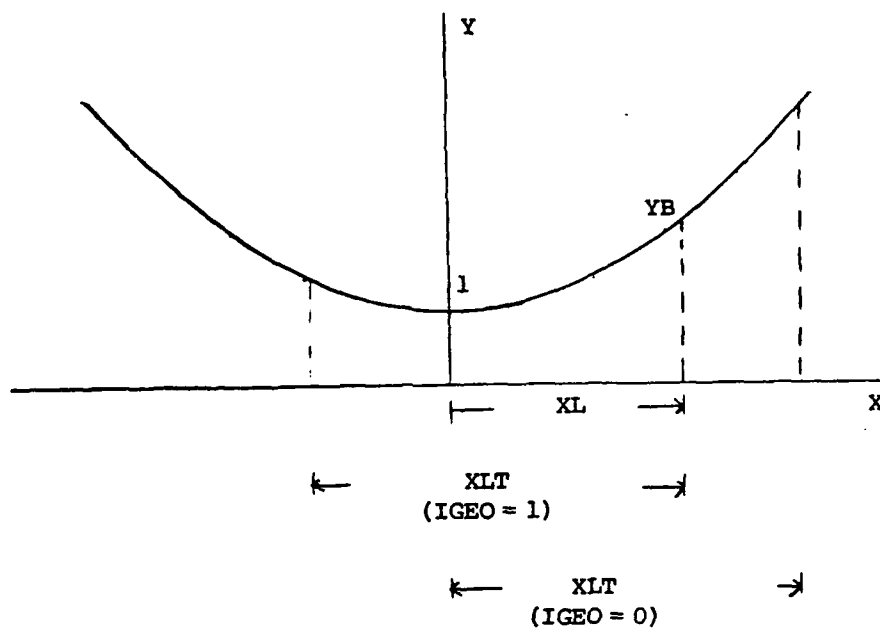


Fig. B 1. Geometry Input Variable

Appendix C

PROGRAM SUBROUTINE DESCRIPTION

The following is a list of subroutines used to solve the nonequilibrium streamtube flow problem. A description of each subroutine is provided below.

MAIN

- executive routine to manage input, output, initialize data, and check convergence.

INTEG

- integration subroutine. Computes explicit and implicit changes in dependent variable U1 and applies cut off criterion if ICTOFF = 1. Called from PROGRAM MAIN.

EVECT

- computes matrix of eigenvectors (SX) and its inverse (SXINV). Called from SUBROUTINE INTEG.

BNDRY

- calculates the characteristic boundary condition. Called from SUBROUTINE INTEG.

CHEM

- calculates the species continuity source term OMEGA. Called from PROGRAM MAIN.

Appendix D

COMPUTER PROGRAM LISTING

The following pages contain the complete listing of the nonequilibrium streamtube flow program described in this paper.

```

PROGRAM MAIN
COMMON U1(61,4),SX(4,4),SXINV(4,4),DU(61,4),U(61,4),
1      A(61),C,CFL,DT,DY,GAM0,II,NMAX,SW,IBUG,FLUXS(4),ICALL
2      ,IPND
3      ,OMEGA(61),TCHTM(61),Q(61),ACHEM,CVF,ETA,ICHEM,PA2,
4      PD,RHO0,RD,THD,TPLX,T0,CE(61),DCDR1,DCDT1,DNP,IEORAT
5      ,ECON(61),ICTOFF,XKCHEM,INLET,Y4IN
DIMENSION X(61),XH(61),CN(61)
1      ,U15(61,4),PXN(61),XMDCT(61)
C-----
C----- INPUT DATA
C-----
      READ(4,200)NMAX,ITMAX,IWRITE,IBUG,DELMAX
      READ(4,210)XL,XLT,YB,CFL
      READ(4,220)PHO1,PHO2,T1,T2,V0,V1
      READ(4,222)ICHEM,TC,PO,XMA2
      READ(4,223)PD,THD,ETA,TPLX,CF
      READ(4,221)ICON,IPT
      READ(4,1221)ICTOFF,IGEO,INLET,IREAD,ITAPE
      READ(4,205)CA1,XKCHEM
      CLOSE(UNIT=4,DISPOSE='SAVE',ERR=999)
      IF(ICTOFF.EQ.0) GO TO 2000
      READ(78,901) (A(I),U(I,1),U(I,2),U(I,3),ECON(I),I=1,NMAX)
      CLOSE(UNIT=78,DISPOSE='SAVE',ERR=999)
2000  CONTINUE
C-----
C----- PRINT INPUT DATA
C-----
      WRITE(61,225)
      WRITE(61,230)NMAX,ITMAX,IWRITE,IBUG,DELMAX
      WRITE(61,240)CFL,XL,XLT,YB
      WRITE(61,250)RHO1,RHO2,T1,T2,V0,V1
      WRITE(61,252)ICHEM,TC,PO,XMA2
      WRITE(61,253)PD,THD,ETA,TPLX,CF
      WRITE(61,251)ICON,IPT
      WRITE(61,254)ICTOFF,IGEO,INLET,IREAD,ITAPE
      WRITE(61,256)CA1,XKCHEM
C-----
C----- NOZZLE GEOMETRY
C-----
      A1 = XLT
      IF(IGEO.EQ.0) A1 = XLT - XL
      DX = A1/FLOAT(NMAX-1)
      A2 = XLT
      IF(IGEO.EQ.0) A2 = 0.
      DO 5 I = 1,NMAX
      X(I) = XL - A2 + DX*FLOAT(I-1)
      A(I) = (YB - 1.0)*X(I)**2/(XL**2) + 1.0
5      CONTINUE
C-----
C----- INITIAL CONDITIONS
C-----
      RU = 8.315
      IEORAT = 0

```

```

TIME = 0.0
DT = 0.0
SW = 0.0
ITER = 0
ISAVE = 1
CFLMIN = 1.0E+06
NMIN = 1
IF(INLET .EQ. 1) NMIN=2
DO 10 I = 1,NMAX
  U(I,1) = RH01 - PH02*(Y(I) - X(1))/ A1
  U(I,3) = T1 - T2*(X(I) - X(1))/ A1
  U(I,2) = V0 + V1*(X(I) - X(1))/ A1
  XM(I) = U(I,2)/SORT(U(I,3))
  U(I,4) = 0.0
  CN(I) = 0.0
  RXN(I) = 0.0
10  CONTINUE
  IF(IREAD .EQ. 0) GO TO 2010
  READ(98,901) (A(I),U(I,1),U(I,2),U(I,3),U(I,4),I=1,NMAX)
  CLOSE(UNIT=98,DISPOSE='SAVE',EPR=999)
2010 CONTINUE
  XM1 = U(NMAX,1)*U(NMAX,2)*A(NMAX)
  DO 11 K = 1,4
    FLUXS(K) = 0.
    DO 11 I = 1,NMAX
      U1(I,K) = U(I,K)
      U1S(I,K) = U(I,K)
      XMDOT(I) = U(I,1)*U(I,2)*A(I)/XM1
11  CONTINUE
  RA2 = RU/XMA2
  RO = RA2
  CVF = RO/(GAMO-1.)
C-----
C----- STAGNATION CHEMISTRY
C-----
  IF(ICHEM .EQ. 1) GO TO 14
  PO = PO*1.0E+05/.9869
  RA = 2.*RU/XMA2
  RA2 = RU/XMA2
  CVF = 3.*RA2
  A1 = 1.0E+06*RA2*T0*PO*EXP(-THD/T0)/PO
  CAO = SORT(A1 + A1**2)/(A1 + 1.)
  RO = (1. + CAO)*RA2
  RH00 = PO/(RO*T0*1.0E+06)
  GAMO = (RO + CVF)/CVF
  RDB = RO/RH00
  A1 = 1. - CAO
  DSTG = (A1**A1)*(CAO**(2.*CAO))
  IF(IREAD .EQ. 1) GO TO 3
  DO 13 I = 1,NMAX
    A2 = THD/(T0*U(I,3))
    IF(A2 .GT. 85.) U(I,4) = 0.0
    IF(A2 .GT. 85.) GO TO 12
    A1 = RO*EXP(-THD/(T0*U(I,3)))/(RH00*U(I,1))

```

IN IV

V02.5

Mon 24-Jan-83 13:46:37

PAGE 003

```

      U(I,4) = (-A1 + SQRT(4.*A1 + A1**2))/(2.)
      IF(ICTOFF .EQ. 1) U(I,4) = CA1
12     U1(I,4) = U(I,4)
      U1S(I,4) = U(I,4)
      R = RA2*(1. + U1(I,4))
      GAM = (P + CVF)/CVF
      XM(I) = U(I,2)/SQRT(GAM*R*U(I,3)/(GAM*RO))
13     CONTINUE
      IF(ICHEM .EQ. 3 .OR. CA1 .EQ. 0.0) GO TO 3
      U(1,4) = CA1
      U1(1,4) = CA1
      U1S(1,4) = CA1
3      CONTINUE
C-----
C----- PRINT INITIAL/BOUNDARY CONDITIONS
C-----
14     WRITE(61,255)
      WRITE(61,260)DX
      IF(ICHEM .NE. 1)WRITE(61,261)RA2,CVF,RH00,RO
      WRITE(61,270) (I,X(I),A(I),I=1,NMAX)
      WRITE(61,400)ITER,TIME,DT,IEQRAT
      WRITE(61,410)
      WRITE(61,420) (I,X(I),U(I,1),U(I,2),U(I,3),XM(I),
1         U(I,4),CN(I),FXN(I),XMDOT(I),I=1,NMAX)
      IF(IBUG .EQ. 0) GO TO 600
      WRITE(61,520)
      WRITE(61,510)(U1(I,1),U1(I,2),U1(I,3),U1(I,4),I=1,NMAX)
520    FORMAT(//,1X,'MAIN INITIAL CONDITIONS U1')
600    CONTINUE
      IF(ICON. EQ. 0) GO TO 300
      WRITE(61,430)IPT
300    CONTINUE
C*****C
C                                     C
C      INTEGRATION LOOP                C
C                                     C
C*****C
      ICOUNT = IWRITE
      ITER = 1
100    CONTINUE
C-----
C----- FINITE RATE CHEMISTRY
C-----
      TMIN = 1.0E+06
      TMIN = XKCHEM*TMIN
C-----
C----- CALCULATE TIME STEP
C-----
59     VMAX = 0.0
      DO 60 I = 1,NMAX
      P = (1. + U1(I,4))*PA2
      A1 = (R**2 + CVF*R)/(GAM*RO*CVF)
      TEST = ABS(U(I,2)) + SQRT(A1*U(I,3))
      IF(TEST .GT. VMAX) VMAX = TEST

```

```

60  CONTINUE
    DT = CFL*DX/VMAX
    IF(ICHEM .NE. 2) GO TO 62
    TFLOW = DX/VMAX
62  DO 61 I = 1,NMAX
    R = (1. + U1(I,4))*RA2
    A1 = (R**2 + CVF*R)/(GAMO*RO*CVF)
    CN(I) = DT*(ABS(U(I,2)) + SQRT(A1*U(I,3)))/DX
    IF(CN(I) .LT. CFLMIN) CFLMIN = CN(I)
    IF(CN(I) .GT. 1.0 .OR. CFL .LT. 1.0) GO TO 63
    WRITE(61,440) ITER,I,CN(I)
63  CONTINUE
61  CONTINUE
    TIME = TIME + DT
C-----
C----- PREDICTOR STEP
C-----
    ICALL = 1
    DO 70 I = 1,NMAX
    A2 = THD/(TO*U1(I,3))
    IF(A2 .GT. 85.) CE(I) = 0.0
    IF(A2 .GT. 85.) GO TO 70
    CE(I) = RO*EXP(-THD/(TO*U1(I,3)))/(RHOO*U1(I,1))
70  CONTINUE
    IF(ICHEM .EQ. 2) CALL CHEM
    CALL INTEG
    DO 15 K = 1,4
    DO 15 I = NMIN,NMAX
    U1(I,K) = U(I,K) + DU(I,K)
15  CONTINUE
C-----
C----- BOUNDARY CONDITIONS
C-----
    IF(INLET .EQ. 1) GO TO 2020
    A1 = RO*EXP(-THD/(TO*U1(1,3)))/(RHOO*U1(1,1))
    IF(ICHEM .NE. 2) U1(1,4) = (-A1 + SQRT(4.*A1 + A1**2))/2.
    GAM = (4. + U1(1,4))/3.
    COEF = GAMO/GAM
    A1 = CAO - U1(1,4)
    A2 = 1. - U1(1,4)
    A3 = U1(1,4)
    A4 = (RDB**A1)*(A3**(2.*A3))*(A2**A2)*EXP(A1)
    COEF = DSTG/A4
    U1(1,1) = (COEF*U1(1,3)**3)**(1./(1.+A3))
    U1(1,2) = SQRT(2.*PA2*(4.+CAO-(4.+A3)*U1(1,3) +
1      A1*THD/TO)/(GAMO*RO))
2020 CONTINUE
    IF(IRUG .EQ. 0) GO TO 610
    WRITE(61,500)
    WRITE(61,510)(U1(I,1),U1(I,2),U1(I,3),U1(I,4),I=1,NMAX)
500  FORMAT(/,1X,'MAIN PREDICTOR U1')
510  FORMAT(1X,4E14.6)
610  CONTINUE
C-----

```

C----- CORRECTOR STEP

C-----

```

      ICALL = 2
      SW = 1. - SW
      DO 76 I = 1,NMAX
      A2 = THD/(TO*U1(I,3))
      IF(A2 .GT. 85.) CE(I) = 0.0
      IF(A2 .GT. 85.) GO TO 76
      CE(I) = RD*EXP(-THD/(TO*U1(I,3)))/(RH00*U1(I,1))
76    CONTINUE
      IF(ICHEM .EQ. 2) CALL CHEM
      CALL INTEG
      DO 24 I = 1,NMAX
      RYN(I) = TCHEM(I)
24    CONTINUE
      DO 25 K = 1,4
      DO 25 I = NMIN,NMAX
      U1(I,K) = 0.5*(U(I,K) + U1(I,K) + DU(I,K))
25    CONTINUE

```

C-----

C----- BOUNDARY CONDITIONS

C-----

```

      IF(INLET .EQ. 1) GO TO 2030
      A1 = RD*EXP(-THD/(TO*U1(1,3)))/(RH00*U1(1,1))
      IF(ICHEM .NE. 2) U1(1,4) = (-A1 + SQRT(4.*A1 + A1**2))/2.
      GAM = (4. + U1(1,4))/3.
      COEF = GAM0/GAM
      A1 = CAO - U1(1,4)
      A2 = 1. - U1(1,4)
      A3 = U1(1,4)
      A4 = (RCP**A1)*(A3**(2.*A3))*(A2**A2)*EXP(A1)
      COEF = DSTG/A4
      U1(1,1) = (COEF*U1(1,3)**3)**(1./(1.+A3))
      U1(1,2) = SQRT(2.*CA2*(4.+CA0-(4.+A3)*U1(1,3) +
1      A1*THD/TO)/(GAM0*RO))
2030 CONTINUE
      IF(IBUG .EQ. 0) GO TO 620
      WRITE(61,530)
530    FORMAT(//,1X,'MAIN CORRECTOR U1')
      WRITE(61,510) (U1(I,1),U1(I,2),U1(I,3),U1(I,4),I=1,NMAX)
620    CONTINUE

```

C-----

C----- PRINT OUTPUT

C-----

```

      IF(ICON .EQ. 0) GO TO 31
      WRITE(61,435) ITER,U1(IPT,1),U1(IPT,2),U1(IPT,3),U1(IPT,4)
      GO TO 35
31    IF(ITER .NE. ICCUNT) GO TO 35
      ICCUNT = ICCUNT + 1
      XM1 = U1(NMAX,1)*U1(NMAX,2)*A(NMAX)
      DO 30 I = 1,NMAX
      IF(ICHEM .EQ. 1) GO TO 33
      P = PA2*(1. + U1(I,4))
      GAM = (P + CVF)/CVF

```

```

      XM(I) = U1(I,2)/SQRT(GAM*P*U1(I,3)/(GAMQ*PO))
      IF(ICHEM.EQ. 2) GO TO 32
      A1 = U1(I,4)
      A2 = TO*U1(I,3)/THO
      A3 = (4.+A1)*(1.+A1)*(A1*(1.-A1)+3.*(2.-A1)*A2**2)
      A4 = 3.*A1*(1.-A1**2)*(1.+2.*A2)
1      + 3.*(8.+3.*A1-A1**3)*A2**2
      A5 = SQRT(A2/A4)
      XM(I) = A5*XM(I)
      GO TO 32
33      CONTINUE
      XM(I) = U1(I,2)/SQRT(U1(I,3))
32      CONTINUE
      XMDOT(I) = U1(I,1)*U1(I,2)*A(I)/XM1
30      CONTINUE
      WRITE(61,400) ITER,TIME,DT,IEQRAT
      WRITE(61,410)
      WRITE(61,420) (I,X(I),U1(I,1),U1(I,2),U1(I,3),XM(I),
1      U1(I,4),CN(I),RXN(I),XMDOT(I),I=1,NMAX)
35      CONTINUE
C-----
C----- CONVERGENCE TEST
C-----
      IF(ITER.NE. 10) GO TO 159
      SUM = 0.0
      DO 149 K = 1,4
      DO 149 I = 1,NMAX
      SUM = SUM + ((U1(I,K) - U1S(I,K))**2)
149      CONTINUE
      SUM1 = SQRT(SUM/FL0AT(NMAX))
159      CONTINUE
      IF(ISAVE.NE. 10) GO TO 41
      ISAVE = 0
      DEL = 0.0
      SUM = 0.0
      DO 40 K = 1,4
      DO 40 I = 1,NMAX
      TEST = ABS(U1(I,K) - U1S(I,K))
      IF(TEST.GT. DEL) DEL = TEST
      SUM = SUM + ((U1(I,K) - U1S(I,K))**2)
40      CONTINUE
      SUM = SQRT(SUM/FL0AT(NMAX))
      SUM = ALOG10(SUM/SUM1)
      IF(ITAPE.EQ. 2) WRITE(77,900) ITER,SUM
      IF(DEL.LT. DELMAX) GO TO 998
      DO 42 K = 1,4
      DO 42 I = 1,NMAX
      U1S(I,K) = U1(I,K)
42      CONTINUE
41      CONTINUE
C-----
C----- RESET ARRAYS TO NEW VALUES
C-----
      ITER = ITER + 1

```



```

      ISAVE = ISAVE + 1
      IF(ITEP .GT. ITMAX) GO TO 999
      DO 45 K = 1,4
      DO 45 I = 1,NMAX
      U(I,K) = U1(I,K)
45    CONTINUE
      GO TO 100
C*****C
C      C
C      END OF LOOP      C
C      C
C*****C
998    XM1 = U1(NMAX,1)*U1(NMAX,2)*A(NMAX)
      DO 55 I = 1,NMAX
      IF(ICHEM .EQ. 1) GO TO 53
      R = RA2*(1. + U1(I,4))
      GAM = (F + CVF)/CVF
      XM(I) = U1(I,2)/SQRT(GAM*R*U1(I,3)/(GAM0*R0))
      IF(ICHEM .EQ. 2) GO TO 54
      A1 = U1(I,4)
      A2 = T0*U1(I,3)/TH0
      A3 = (4.+A1)*(1.+A1)*(A1*(1.-A1)+3.*(2.-A1)*A2**2)
      A4 = 3.*A1*(1.-A1**2)*(1.+2.*A2)
1      + 3.*(P.+3.*A1-A1**3)*A2**2
      A5 = SQRT(A3/A4)
      XM(I) = A5*XM(I)
      GO TO 54
53    CONTINUE
      XM(I) = U1(I,2)/SQRT(U1(I,3))
54    CONTINUE
      XMDOT(I) = U1(I,1)*U1(I,2)*A(I)/XM1
55    CONTINUE
      WRITE(61,400) ITER,TIME,DT,IEQRAT
      WRITE(61,410)
      WRITE(61,420) (I,X(I),U1(I,1),U1(I,2),U1(I,3),XM(I),
1      U1(I,4),CN(I),RXN(I),XMDOT(I),I=1,NMAX)
999    CONTINUE
      WRITE(61,445) CFLMIN
      CLOSE(UNIT=61,DISPOSE='SAVE')
      IF(ITAPE .EQ. 1)WRITE(99,901)(A(I),U1(I,1),U1(I,2),U1(I,3),
1      U1(I,4),I=1,NMAX)
      STOP
C-----
C----- FORMAT STATEMENTS
C-----
200    FORMAT(4I5,F10.0)
205    FORMAT(2F10.0)
210    FORMAT(4F10.0)
220    FORMAT(6F10.0)
1221   FORMAT(5I5)
221    FORMAT(2I5)
222    FORMAT(I5,3F10.0)
223    FORMAT(3F10.0,2E10.3)
224    FORMAT(4E14.6)

```

AN IV

V02.5

Mon 24-Jun-83 13:46:37

PAGE 008

```

225  FORMAT(//,1X,'INPUT DATA',//)
230  FORMAT(1X,'NMAX=',I7,3X,'ITMAX=',I5,3X,'IWRITE=',I5,3X,
1    'IPUG=',I5,3X,'DELMAX=',E10.3)
240  FORMAT(/,1X,'CFL=',F10.3,3X,'XL=',E10.3,3X,
1    'XLT=',E10.3,3X,'Y2=',E10.3)
250  FORMAT(/,1X,'FHC1=',F18.10,3X,'RHO2=',E10.3,3X,'T1=',E18.10,3X,
1    'T2=',E10.3,3X,'VO=',E18.10,3X,'V1=',E10.3)
251  FORMAT(/,1X,'ICON=',I5,3X,'IPT=',I5)
254  FORMAT(/,1X,'ICTOFF=',I5,3X,'IGEO=',I5,3X,'INLET=',I5,3X,
1    'IREAD=',I5,3X,'ITAPE=',I5)
252  FORMAT(/,1X,'ICHTH=',I5,3X,'TO=',E10.3,3X,'PO=',E10.3,
1    3X,'XMA2=',F10.3)
253  FORMAT(/,1X,'RD=',E10.3,3X,'THD=',E10.3,3X,'ETA=',E10.3,
1    3X,'TPLX=',E10.3,3X,'CF=',E10.3)
255  FORMAT(//,'INITIAL/BOUNDARY CONDITIONS',//)
256  FORMAT(/,1X,'CA1=',E10.3,3X,
1    'XKCHEM=',E10.3)
260  FORMAT(1X,'DX=',E10.3,/)
261  FORMAT(/,1X,'RA2=',F18.10,3X,'CVF=',E18.10,3X,
1    'RHO0=',E18.10,3X,'RO=',E18.10)
270  FORMAT(1X,'I=',I3,3X,'X=',E14.6,3X,'A=',E14.6)
400  FORMAT(//,1X,'ITER=',I4,3X,'TIME=',E14.6,3X,'DT=',E14.6
1    3X,'IEQPAT=',I4)
410  FORMAT(/,4X,'I',8X,'X',12X,'RHO',12X,'U',13X,'T',13X,'M',12X,
1    'CA',12X,'CH',12X,'RXN',10X,'XMDOT')
420  FORMAT(3X,I3,9E14.6)
430  FORMAT(//,1X,'CONVERGENCE CHECK-U1,U2,U3',3X,'I=',I5,//)
435  FORMAT(1X,I5,3X,4F14.6)
440  FORMAT(//,1X,'ITER=',I4,3X,'I=',I4,3X,'CN=',E14.6,//)
445  FORMAT(//,1X,'CFLMIN=',E14.6)
441  FORMAT(//,1X,'ITER=',I4,3X,'I=',I4,3X,'RXN=',E14.6,//)
900  FORMAT(1X,I5,E14.6)
901  FORMAT(5F14.6)

```

END

```

SUBROUTINE INTEG
COMMON U1(61,4),SX(4,4),SXINV(4,4),DU(61,4),U(61,4),A(61),
1      C,CFL,DT,DX,GAM0,II,NMAX,SW,ISUG,FLUXS(4),ICALL
2      ,TEND
3      ,OMEGA(61),TCHEM(61),Z(61),ACHEM,CVF,ETA,ICHEM
4      ,PA2,PD,RH00,PC,TH0,TRLX,TC,CE(61),DCDR1,DCDT1,DMP,IEOP
5      ,ECON(61),ICTOFF,XKCHEM,INLET,NMIN
DIMENSION DA(4),DAINV(4),FLUX(4),W(4),X(4),Y(4),Z(4)
C-----
C----- EXPLICIT STAGE
C-----
      IF(ICTOFF .EQ. 0) GO TO 2000
      IEORAT = 0
      DO 80 I = 2,NMAX
      IF(I .NE. NMAX) GO TO 71
      C4 = U1(I,4) - U1(I-1,4)
      GO TO 72
71      C4 = (1.-SW)*U1(I+1,4)-(1.-2.*SW)*U1(I,4)-SW*U1(I-1,4)
72      DU(I,4) = -DT*(U1(I,2)*C4)/DX + DT*OMEGA(I)
      DEQ = ECON(I) - U1(I,4)
      IF(ABS(DU(I,4)) .GE. ABS(DEQ)) IEORAT = I
80      CONTINUE
2000     CONTINUE
      ISND = 0
      DCDR1 = 0.0
      DCDT1 = 0.0
      DO 5 I = NMIN,NMAX
      IF(I .NE. 1) GO TO 2
      C1 = U1(I+1,1) - U1(I,1)
      C2 = U1(I+1,2) - U1(I,2)
      C3 = U1(I+1,3) - U1(I,3)
      C4 = U1(I+1,4) - U1(I,4)
      C5 = ALOG(A(I+1)) - ALOG(A(I))
      GO TO 4
2      IF(I .NE. NMAX) GO TO 3
      C1 = U1(I,1) - U1(I-1,1)
      C2 = U1(I,2) - U1(I-1,2)
      C3 = U1(I,3) - U1(I-1,3)
      C4 = U1(I,4) - U1(I-1,4)
      C5 = ALOG(A(I)) - ALOG(A(I-1))
      GO TO 4
3      CONTINUE
      C1 = (1.-SW)*U1(I+1,1) - (1.-2.*SW)*U1(I,1) - SW*U1(I-1,1)
      C2 = (1.-SW)*U1(I+1,2) - (1.-2.*SW)*U1(I,2) - SW*U1(I-1,2)
      C3 = (1.-SW)*U1(I+1,3) - (1.-2.*SW)*U1(I,3) - SW*U1(I-1,3)
      C4 = (1.-SW)*U1(I+1,4) - (1.-2.*SW)*U1(I,4) - SW*U1(I-1,4)
      C5 = (1.-SW)*ALOG(A(I+1)) - (1.-2.*SW)*ALOG(A(I))
      1      - SW*ALOG(A(I-1))
4      CONTINUE
      DCDR = 0.0
      DCDT = 0.0
      DU(I,4) = 0.0
      IF(ICHEM .EQ. 3) GO TO 54
      IF(ICHEM .EQ. 1) OMEGA(I) = 0.0

```

```

DU(I,4) = -DT*(U1(I,2)*C4)/DX + DT*OMEGA(I)
ICUT = 0
IF(ICTOFF .EQ. 0) GO TO 55
DEQ = XKCHEM*(ECON(I) - U1(I,4))
CON = U1(I,4)**2/(1.-U1(I,4))
CONP = U1(I,4) + DU(I,4)
EQCRIT = U1(I,4) + DEQ
IF(CONP .LE. 0.0 .OR. CONP .GE. 1.0) GO TO 58
EQRAT = CON/CE(I)
IF(I .GT. IEQRAT) GO TO 55
58 ICUT = 1
DU(I,4) = DEQ
OMEGA(I) = DU(I,4)/DT + U1(I,2)*C4/DX
Q(I) = OMEGA(I)*THD/(3.*T0)
55 IF(ICALL .EQ. 2) GO TO 56
TCHEM(I) = 1.0
IF(ICUT .EQ. 1) TCHEM(I) = 3.
GO TO 60
56 IF(ICUT .EQ. 0) GO TO 60
IF(TCHEM(I) .EQ. 3.0) GO TO 57
TCHEM(I) = 2.0
GO TO 60
57 TCHEM(I) = 4.0
GO TO 60

C-----
C----- EQUILIBRIUM
C-----
54 A1 = CF(I)
A2 = U1(I,4)
U1(I,4) = (-A1 + SQRT(4.*A1 + A1**2))/2.
DU(I,4) = U1(I,4) - A2
OMEGA(I) = 0.0
Q(I) = 0.0
C4 = 0.0
A1 = U1(I,4)
A2 = THD/T0
A3 = A1*(1.-A1)/(2.-A1)
DCDR = -A3/U1(I,1)
DCDT = A2*A3/(U1(I,3)**2)
IF(I .EQ. 1) DCDR1 = DCDR
IF(I .EQ. 1) DCDT1 = DCDT
TCHEM(I) = 4.0

C-----
C----- NONEQUILIBRIUM
C-----
60 CONTINUE
R = RA2*(1. + U1(I,4))
B1 = 1./(1. + THD*DCDT/(3.*T0))
B2 = THD*DCDR/(3.*T0)
DU(I,1) = -DT*(U1(I,2)*C1 + U1(I,1)*C2 + U1(I,1)*U1(I,2)*C5)/DX
DU(I,2) = -DT*U1(I,2)*C2/DX - DT*(R*C3+RA2*U1(I,3)*(C4
1 + DCDR*C1 + DCDT*C3)
2 + P*U1(I,3)*C1/U1(I,1))/(GAMO*RO*DX)
IF(ICHEM .EQ. 1) Q(I) = 0.0

```

```

      DU(I,3) = -DT*U1(I,2)*C3/DX - DT*B1*(P*U1(I,3)/CVF - P2*U1(I,1)
1      *(C2 + U1(I,2)*C5)/DX - DT*Q(I)
5      CONTINUE
      DU(I,4) = 0.0
      IF(ISUG.EQ. 0) GO TO 200
      WRITE(61,100)
100     FORMAT(/,1X,'INTEG EXPLICIT DU')
      WRITE(61,110)(DU(I,1),DU(I,2),DU(I,3),DU(I,4),OMEGA(I),CE(I),
1      I=1,NMAX)
110     FORMAT(1X,6E14.6)
200     CONTINUE
      IF(INLET.EQ. 0) CALL ENDRY
C-----
C----- STABILITY CHECK
C-----
      IF(CFL.LE. 1.0) RETURN
C-----
C----- BOUNDARY CONDITIONS
C-----
      DO 47 K = 1,4
      FLUX(K) = FLUXS(K)
47      CONTINUE
      ISAVE = 2
      IF(INLET.EQ. 0) GO TO 2010
      ISAVE = 0
      DO 410 K = 1,4
      FLUXS(K) = 0.0
410     CONTINUE
2010     CONTINUE
      IF(SW.EQ. 1.0 .AND. ICALL.EQ. 1) ISAVE = NMAX-1
      IF(SW.EQ. 0.0 .AND. ICALL.EQ. 2) ISAVE = NMAX-1
C-----
C----- IMPLICIT STAGE
C-----
      IBND = 1
      DO 40 N = NMIN,NMAX
      I = SW*N + (1.-SW)*(NMAX+NMIN-N)
      II = I
      IF(I.NE. 1) GO TO 13
      CALL ENDRY
      DO 11 K = 1,4
      X(K) = 0.0
      DO 11 L = 1,4
      X(K) = Y(K) + SX(K,L)*FLUX(L)
11      CONTINUE
      DO 12 K = 1,4
      FLUX(K) = X(K)
12      CONTINUE
13      CONTINUE
      CALL EVCT
      DO 10 K = 1,4
      W(K) = DU(I,K) + DT*FLUX(K)/DX
10      CONTINUE
      DO 15 K = 1,4

```

```

X(K) = 0.0
DO 15 L = 1,4
X(K) = X(K) + SX(K,L)*W(L)
15  CONTINUE
DA(1) = ABS(U1(I,2) - ACHEM*C) - (DX/DT)
DA(2) = ABS(U1(I,2)) - (DX/DT)
DA(4) = ABS(U1(I,2)) - (DX/DT)
DA(3) = ABS(U1(I,2) + ACHEM*C) - (DX/DT)
IF(ICHEM .NE. 2 .OR. I .EQ. 1) DA(4) = 0.0
DO 20 K = 1,4
DA(K) = AMAX1(DA(K),0.0)
DAINV(K) = 1./(1. + DT*DA(K)/DX)
IF(ICHEM .NE. 2 .OR. I .EQ. 1) DAINV(4) = 0.0
Y(K) = DAINV(K)*X(K)
20  CONTINUE
IF(I .NE. 1) GO TO 24
A1 = DAINV(1)
A2 = ((U1(I,2) - ACHEM*C)/(U1(I,2) + ACHEM*C))*DA(1)*A1*DT/DX
Y(1) = A1*X(1)
Y(2) = X(2)
Y(3) = X(3) - A2*X(1)
Y(4) = 0.0
24  DO 25 K = 1,4
DU(I,K) = 0.0
DO 25 L = 1,4
DU(I,K) = DU(I,K) + SXINV(K,L)*Y(L)
25  CONTINUE
DO 30 K = 1,4
Z(K) = DA(K)*Y(K)
30  CONTINUE
DO 35 K = 1,4
FLUX(K) = 0.0
DO 35 L = 1,4
FLUX(K) = FLUX(K) + SXINV(K,L)*Z(L)
35  CONTINUE
IF(I .NE. ISAVE) GO TO 37
DO 38 K = 1,4
FLUXS(K) = FLUX(K)
38  CONTINUE
37  CONTINUE
IF(IBUG .EQ. 0) GO TO 210
WRITE(61,120)I,DX,DT
120  FORMAT(//,1X,'INTEC IMPLICIT STEP',3X,'I=',I3,3X,'DX=',
1      F7.4,3X,'DT=',F7.4,/,1X,'W,X,Y,Z,DU,FLUX')
WRITE(61,130)(W(K),X(K),Y(K),Z(K),DU(I,K),FLUX(K),K=1,4)
130  FORMAT(1X,6E14.6)
210  CONTINUE
40  CONTINUE
RETURN
END

```

```

SUBROUTINE ETECT
COMMON U1(61,4),SX(4,4),SXINV(4,4),DU(61,4),U(61,4),
1      A(61),C,CFL,DT,DX,GAM0,II,NMAX,SW,IRUG,FLUXS(4),ICALL
2      ,IFND
3      ,OMEGA(61),TCHEN(61),T(61),ACHEM,CVF,ETA,ICHEM,
4      RA2,RO,RHCO,PO,THD,TRLX,TO,CE(61),DCDR1,DCDT1,DMP,IEC
5      ,FCON(61),ICTOFF,XKCHEM

I = II
C = SORT(U1(I,3))
R = RA2*(1. + U1(I,4))
DCDR = 0.0
DCDT = 0.0
ACHEM = SORT((CVF*R + R**2)/(GAM0*RO*CVF))
IF(ICHEM.NE. 3) GO TO 2
A1 = U1(I,4)
A2 = TO*U1(I,3)/THD
A3 = GAM0*RO*(A1*(1.-A1)+3.*(2.-A1)*A2**2)
A4 = A1*(1.-A1**2)*(1.+2.*A2)+(8.+3.*A1-A1**3)*A2**2
ACHEM = SORT(RA2*A4/A3)
B1 = U1(I,4)*(1.-U1(I,4))/(2.-U1(I,4))
B2 = THD/TO
DCDR = -B1/U1(I,1)
DCDT = B2*B1/(U1(I,3)**2)
2  CONTINUE
A1 = 1./(1. + THD*DCDT/(3.*TO))
A2 = THD*DCDR/(3.*TO)
IF(ICHEM.EQ. 1) ACHEM = 1.
IF(IRUG.EQ. 0) GO TO 200
WRITE(61,100)I,ACHEM,C
100  FORMAT(/,1X,'ETECT',3X,'I=',I3,3X,'A=',E14.6,3X,'C=',E14.6)
200  CONTINUE

C-----
C----- CALCULATE MATRIX OF EIGENVECTORS
C-----
SX(1,4) = 0.
SX(2,4) = 0.
SX(3,4) = 0.
SX(4,1) = 0.
SX(4,2) = 0.
SX(4,3) = 0.
SX(4,4) = 0.
SX(1,1) = 1.
SX(1,2) = -ACHEM*C*GAM0*RO*U1(I,1)/(R*U1(I,3)+
1      RA2*U1(I,3)*U1(I,1)*DCDR)
SX(1,3) = (U1(I,1)/U1(I,3))
1      *(R+RA2*U1(I,3)*DCDT)/(R+RA2*U1(I,1)*DCDR)
SX(2,1) = 1.
SX(2,2) = 0.
SX(2,3) = -U1(I,1)/(A1*(R*U1(I,3)/CVF - A2*U1(I,1)))
SX(3,1) = 1.
SX(3,2) = -SX(1,2)
SX(3,3) = SX(1,3)
SX(4,1) = 0.
SX(4,2) = 0.

```

```

SX(4,3) = 0.
IF(ICHEM .NE. 2 .OR. I .EQ. 1) GO TO 10
SX(1,4) = PA2*U1(I,1)/R
SX(2,4) = 0.
SX(3,4) = SX(1,4)
SX(4,4) = 1.

```

C-----

C----- CALCULATE INVERSE MATRIX

C-----

```

10  A1 = SX(3,2)*SX(1,3)-SX(1,2)*SX(3,3)
1    -SX(2,3)*(SX(3,2)-SX(1,2))
    SXINV(1,4) = 0.
    SXINV(2,4) = 0.
    SXINV(3,4) = 0.
    SXINV(4,1) = 0.
    SXINV(4,2) = 0.
    SXINV(4,3) = 0.
    SXINV(4,4) = 0.
    SXINV(1,1) = -SX(3,2)*SX(2,3)/A1
    SXINV(1,2) = (SX(3,2)*SX(1,3)-SX(1,2)*SX(3,3))/A1
    SXINV(1,3) = SX(1,2)*SX(2,3)/A1
    SXINV(2,1) = (SX(2,3) - SX(3,3))/A1
    SXINV(2,2) = 0.
    SXINV(2,3) = -SXINV(2,1)
    SXINV(3,1) = SX(3,2)/A1
    SXINV(3,2) = (SX(1,2) - SX(3,2))/A1
    SXINV(3,3) = SXINV(2,1)
    IF(ICHEM .NE. 2 .OR. I .EQ. 1) GO TO 20
    A2 = 1./(R + CVF)
    SXINV(1,4) = -A2*PA2*CVF*U1(I,1)/R
    SXINV(2,4) = 0.
    SXINV(3,4) = -A2*PA2*U1(I,3)
    SXINV(4,1) = 0.
    SXINV(4,2) = 0.
    SXINV(4,3) = 0.
    SXINV(4,4) = 1.
20  CONTINUE
    IF(1BUG .EQ. 0) GO TO 210
    WRITE(61,110)
110  FORMAT(//,1X,'EVECT SX,SXINV')
    WRITE(61,120)(SX(L,1),SX(L,2),SX(L,3),SX(L,4),SXINV(L,1),
1      SXINV(L,2),SXINV(L,3),SXINV(L,4),L=1,4)
120  FORMAT(1X,8E14.6)
210  CONTINUE
    RETURN
    END

```


SUBROUTINE BNDRY

COMMON U1(61,4),SX(4,4),SXINV(4,4),DU(61,4),U(61,4),

1 A(61),C,CFL,DT,DX,CAMU,II,NMAX,SW,IBUG,FLUXS(4),ICALL

2 ,IFND

3 ,OMEGA(61),TCHFM(61),T(61),ACHEM,CVF,ETA,ICHEM,

4 RA2,RO,PHOO,PC,THD,TPLY,T0,CE(61),DCDR1,DCDT1,DMP,IEGFA

5 ,ECOR(61),ICTOFF,XKCHEM

DIMENSION DU1(4)

C-----

C----- CALCULATE P1P2 MATRIX

C-----

R = (1. + U1(1,4))*FA2

Z = U1(1,4)

A1 = SQRT((P**2 + CVF*P)/(GAMO*RO*CVF))

IF(ICHEM.NE.3) GO TO 2

B1 = U1(1,4)

B2 = T0*U1(1,2)/THD

B3 = GAMO*RO*(B1*(1.-B1)+3.*(2.-B1)*B2**2)

B4 = B1*(1.-B1**2)*(1.+2.*B2)+

1 (8.+3.*B1-B1**3)*B2**2

A1 = SQRT(RA2*B4/B3)

2 CONTINUE

A2 = (U1(1,3) + THD/T0)*DCDR1

A3 = (U1(1,3) + THD/T0)*DCDT1

A4 = GAMO*RO/RA2

A5 = THD*DCDR1/(T0*U1(1,3))

A6 = THD*DCDT1/(T0*U1(1,3))

A7 = A1*GAMO*RO*U1(1,1)*SQRT(U1(1,3))/(R*U1(1,3))

1 + RA2*U1(1,3)*U1(1,1)*DCDR1)

A9=(U1(1,1)/U1(1,3))*(P+RA2*U1(1,3)*DCDT1)/(R+RA2*U1(1,1)*DCDR1)

A8 = (-(1.+Z)/U1(1,1)+A5)*(-A7*(4.+Z+A3) - A4*U1(1,2)*A9)

1 + (3./U1(1,3)+A6)*(A4*U1(1,2)

2 +A2*A7)

P11 = A4*U1(1,2)*(3./U1(1,3)+A6)

P12 = -A2*(3./U1(1,3)+A6)+(-(1.+Z)/U1(1,1)+A5)*

1 (4.+Z+A3)

P13 = -A4*U1(1,2)*(-(1.+Z)/U1(1,1)+A5)

SX(1,1) = P11/A8

SX(1,2) = -P11*A7/A8

SX(1,3) = P11*A9/A8

SX(1,4) = 0.0

SX(2,1) = P12/A8

SX(2,2) = -P12*A7/A8

SX(2,3) = P12*A9/A8

SX(2,4) = 0.0

SX(3,1) = P13/A8

SX(3,2) = -P13*A7/A8

SX(3,3) = P13*A9/A8

SX(3,4) = 0.

SX(4,1) = 0.

SX(4,2) = 0.

SX(4,3) = 0.

SX(4,4) = 0.

IF(IBND.EQ.1)RETURN

N IV

V02.5

Mon 24-Jan-83 13:49:53

PAGE 002

C-----

C----- CALCULATE DU(1,K)

C-----

```
      DO 10 K = 1,4
      DU1(K) = 0.
      DO 10 L = 1,4
      DU1(K) = DU1(K) + SX(K,L)*DU(1,L)
10    CONTINUE
      DO 20 K = 1,4
      DU(1,K) = DU1(K)
20    CONTINUE
      IF (IBUG .EQ. 0) GO TO 200
      WRITE(61,100)
      WRITE(61,110) (SX(K,1),SX(K,2),SX(K,3),SX(K,4),DU(1,K),K=1,4)
      WRITE(61,120) A1,A2,A3,A4,A5,A6,A7,A8,A9
100   FORMAT(/,1X,'MEMORY P1P2,DU1')
110   FORMAT(1X,5E14.6)
120   FORMAT(/,1X,9E14.6)
200   CONTINUE
      RETURN
      END
```

V IV

V02.5

Mon 24-Jan-83 13:50:30

PAGE 001

SUBROUTINE CHEM

COMMON U1(61,4),SX(4,4),SXINV(4,4),DU(61,4),U(61,4),

1 A(61),C,CFL,DT,DX,GAMO,II,NMAX,SW,IBUG,FLUXS(4),ICALL,

2 IPAD,OMEGA(61),TCHEM(61),Q(61),ACHEM,CVF,ETA,ICHEM,

3 RA2,RC,RHOC,PO,THD,TRLX,TO,CE(61),DCDR1,DCDT1,DMP,IEGSA

5 ,ECON(61),ICTOFF,XKCHEM

C-----

C----- CALCULATE CHEMICAL PRODUCTION RATE

C-----

DO 10 I = 1,NMAX

IF(CE(I) .NE. 0.0) GO TO 2

OMEGA(I) = 0.0

Q(I) = 0.0

GO TO 10

2 A1 = EXP(-THD/(TO*U1(I,3)))

A2 = U1(I,4)**2/(1.-U1(I,4))

A3 = A2/CE(I)

OMEGA(I) = (1.-A3)*(1.-U1(I,4))*A1*TRLX*U1(I,1)*U1(I,3)**ETA

Q(I) = OMEGA(I) *THD/(3.*TO)

10 CONTINUE

IF(IBUG .EQ. 0) GO TO 300

WRITE(61,200)

WRITE(61,210)(U1(I,1),U1(I,3),U1(I,4),OMEGA(I),I=1,NMAX)

200 FORMAT(//,'CHEM OMEGA,TCHEM')

210 FORMAT(1X,4F18.10)

220 FORMAT(1X,I4,2E14.6)

300 CONTINUE

RETURN

END

AT
LM